National Library
of Canada

Bibliothèque nationale
du Canada

Canadian Theses Service

Service des thèses canadiennes

Ottawa, Canada
K1A 0N4

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents.

NL-339 (r. 88/04) c

Canada

UNIVERSITY OF ALBERTA

Numerical Simulation of Thermal, Multiphase Fluid Flow in an Elastoplastic Deforming Oil Reservoir

by

W. Simon Tortike

A THESIS
SUBMITTED TO THE FACULTY OF GRADUATE STUDIES AND RESEARCH
IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF

Doctor of Philosophy

IN

Petroleum Engineering

Department of Mining, Metallurgical and Petroleum Engineering

Edmonton, Alberta
Fall 1991

National Library of Canada
Canadian Theses Service
Ottawa, Canada
K1A 0N4

Bibliothèque nationale du Canada
Service des thèses canadiennes

The author has granted an irrevocable non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of his/her thesis by any means and in any form or format, making this thesis available to interested persons.

The author retains ownership of the copyright in his/her thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without his/her permission.

L'auteur a accordé une licence irrévocable et non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de sa thèse de quelque manière et sous quelque forme que ce soit pour mettre des exemplaires de cette thèse à la disposition des personnes intéressées.

L'auteur conserve la propriété du droit d'auteur qui protège sa thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

ISBN 0-315-70290-7

Canada

THE UNIVERSITY OF ALBERTA

RELEASE FORM

NAME OF AUTHOR W. Simon Tortike

TITLE OF THESIS Numerical Simulation of Thermal, Multiphase Fluid Flow in an Elastoplastic Deforming Oil Reservoir

DEGREE FOR WHICH THESIS WAS PRESENTED Doctor of Philosophy

YEAR THIS DEGREE GRANTED Fall 1991

Permission is hereby granted to THE UNIVERSITY OF ALBERTA LIBRARY to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only.

The author reserves other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without the author's written permission.

(SIGNED) WS Tortike

PERMANENT ADDRESS:

11106 - 76 Avenue
Edmonton, Alberta
CANADA T6G 0J8

DATED 10th October 1991

©1991 W. Simon Tortike.
All rights reserved.

I often say that when you can measure what you are speaking about,
and express it in numbers, you know something about it,
but when you cannot measure it, when you cannot express it in numbers,
your knowledge is of a meagre and unsatisfactory kind.

— LORD KELVIN, lecture to the Institution of Civil Engineers (1883)

O dear Ophelia!
I am ill at these numbers:
I have not art to reckon my groans.

— Hamlet (Act II, SC. 2, Line 120)

THE UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research, for acceptance, a thesis entitled Numerical Simulation of Thermal, Multiphase Fluid Flow in an Elastoplastic Deforming Oil Reservoir submitted by W. Simon Tortike in partial fulfilment of the requirements for the degree of Doctor of Philosophy in Petroleum Engineering.

Professor S.M. Farouq Ali

Professor K. Barron

Ramon Bentsen

Professor R.G. Bentsen

J. D. Scott

Professor J.D. Scott

JM Whiting

Professor J.M. Whiting

P K W Vinsome

Dr. P.K.W. Vinsome

Date Sept 4, 1991

ABSTRACT

Bitumen and heavy oil resources in Alberta are located in unconsolidated sediments displaying the characteristics of cohesionless soils. Cyclic steam stimulation in these deposits leads to hydraulic fractures and material dilation or contraction.

Recognizing the nonlinear nature of the formation deformation, a project was carried out to develop a coupled three-dimensional thermal numerical model including elasto-plastic geomechanics. Two models were eventually implemented. A fully finite element model, the natural choice, was unable to solve satisfactorily the complex nature of thermal multiphase flow in a porous medium. A coupled finite element deformation model and finite difference flow model was successful in modelling flows, heat transfer, and formation deformation.

The coupled finite difference/finite element model was run for different cases of grid size and injected steam slug size, to observe the different material responses. It was found that lower degrees of coupling provide useful information at considerably lower computing cost.

The results indicate that the material disturbances should be included to model oil recovery processes in these unconsolidated sands.

ACKNOWLEDGEMENTS

The author wishes to express sincere appreciation for the keen interest, guidance, encouragement and generosity shown by his mentor and supervisor, Dr. S.M. Farouq Ali. Dr. Farouq strongly supported this work by ensuring funding and many other resources, fostering the author's forays into numerical analysis, and making available his extraordinarily broad and diverse experience whenever necessary.

The author is indebted to his wife, Susan, for her support, encouragement, and tolerance of the invariably odd working habits of a graduate student.

The study could not have been completed without the detailed discussions with colleagues in the Department of Civil Engineering, in particular, Drs. N. Yoshida, R. Wan, and T.T. Wong. Their critical debate and advice were invaluable. Their company made the author's period as a graduate student most enjoyable.

Dr. R.E. Hayes of the Department of Chemical Engineering is thanked for the use of his IBM RS/6000 workstation.

Finally, the Alberta Oil Sands and Technology Research Authority generously supported this project, for which the agency is gratefully thanked.

TABLE OF CONTENTS

Page

List of Tables xiii

List of Figures xiv

NOMENCLATURE xxii

Chapter I: Introduction 1

Chapter II: Review of Literature 3

- Application of finite elements to multiphase flow problems 3
- Description of the geomechanical behaviour of oil sands 8
- The inclusion of geomechanical behaviour in reservoir models 16
- Comments 22

Chapter III: Statement of the Problem 24

- Statement of the problem 24

Chapter IV: Derivation of the Mathematical Model 25

- Flow equation development: constitutive equations for heat and mass transfer 25
- Derivation of the solid equations and coupling 30
- Parametrization of fluid and rock properties 36

Chapter V: Discretization and Solution of the Equations — Coupled Mathematical Model for Finite Elements 44

- Spatial integration of a general non-linear partial differential equation 44
- General numerical solution of the equations 48
- Application of the method of weighted residuals and expansion by trial functions 48
- The element equation 54
- Structural stiffness matrix 62

A general time integration method for differential equations using finite elements 62
Application of the Newton-Raphson method 68
Algebraic reduction of the derivatives 69
Implementation of the discretized equations as a finite element computer program 73
Discussion 73

Chapter VI: Finite Difference - Finite Element Approach to the Coupled Model 75
A finite difference thermal fluid-solid coupled model 75
Solid terms in the heat equation 77
Solid terms in the fluid flow equations 80
Evaluation of the solid terms 81
A finite element model for evaluating elastoplastic deformations 87
Coupling schemes between the models 96
Description of subroutines in coupled finite-difference and finite element models 101

Chapter VII: Preparation and Analysis of Data: Pre- and Post-Processing 110
Using the pre-processing routines 110
grid 111
dvar 113
create_conns 114
Using the Post Processing Routines 115
postp 116
conprep 119
conslice 128
fp2hdf 131
Image3 131
Summary 132

Chapter VIII : Application of the Models 133
    Verification of the solid model 133
    Choice of material properties and stress distribution 135
    Fluid model discretizations and operating conditions 141
    Results for 3 × 3 × 3 fluid domain with high injection rate 148
    Results for 5 × 5 × 3 fluid domain with high injection rate 158
    Results for 5 × 5 × 3 fluid domain with low injection rate 167
    Results for 3 × 3 × 3 fluid domain with high injection rate and staggered coupling 175
    Results for 3 × 3 × 3 fluid domain with low injection rate and full coupling 184
    Discussion 192
    Comparison of methods 192
    Making the coupled runs 199
    Stress reduction calculations 200
    Stress sampling points 200
    Usefulness of the runs less than fully coupled 201
    Computer time required to make runs 201
    Assumptions and limitations inherent in the models 202
    Validation of the models 203

Chapter IX : Conclusions 205

REFERENCES 206

Appendix A : Coupled Flow and Solid Equations 217
    Water equation 218
    Bitumen equation 219
    Gas equation 220
    Energy equation 222

Appendix B: Statistical Analyses of Rock and Fluid Properties 224
Correlations for saturated steam properties 224

Appendix C: The Method of Weighted Residuals and Finite Elements 246
Approximation of functions 246
Method of weighted residuals 246
Approximation to the solution of linear partial differential equations 247
The finite element method and piecewise-defined trial functions 249
Mapping and isoparametric elements 250
Upwinding 253
Shape and weighting functions of trilinear brick elements 256
Coordinate transformation and the jacobian matrix 261
Gaussian quadrature 262
Triangular Prism Elements 266
Asymmetric Weighting Functions for the Triangular Prism Elements 270

Appendix D: Development of the Fully Coupled Finite Element Program FESPS 273
Overview of the model 273
Description of the modularity of the program 276
Integration module 276
The well model 282
Outline of well calculations 285
Forming Jacobian derivatives: implementation 293
Forming the Jacobian matrix: pseudocode 293
Forming the global equations 296
Global mesh numbering and the RENUM preprocessor 299
Vector arithmetic 302

Vectorized linear solver DSTAIR 305

Auxiliary equations 306

Implementation of pseudo-descriptors 309

Pseudo-descriptor subroutines 311

Cyber FORTRAN200 functions and their emulations 313

Appendix E: A Highly-Implicit Finite Difference Steam Model (Dead Oil) SIM3D3P 317

Expansion of the equations (accumulation terms) 317

Energy balance: 319

Molar balance for water 324

Molar balance for oil 326

Water-steam equilibrium ratio 326

Heat losses to overburden 327

Solution of the equations 328

Adding solid coupling terms to the formulation 329

Appendix F: Equations defining solid model for finite element analysis 331

Geomechanics conventions 331

Elastoplastic numerical formulation 338

Evaluation of key quantities in elastoplastic formulation 347

Incorporating loads into the formulation 353

Appendix G: Sample data for use with PLAST-PT3 361

Appendix H: Sample data for use with SIM3D3P 403

Appendix I: User Manual for FESPS 408

Preprocessor RENUM 408

Symbolic data set for RENUM 411

Main program FESPS 412

Symbolic data set for FESPS 426

LIST OF TABLES

Table | Page
7–1. Source files and routines constituting CONPREP 120
7–2. Selecting data from the IPLAST and SPLAST files 125
7–3. Source files and routines constituting CONSLICE 129
8–1. General properties of formation fluids and rock matrix 135
8–2. General properties of produced fluids 136
8–3. Data for temperature dependent endpoints 136
8–4. Relative permeability data 137
8–5. General rock and fluid properties 137
8–6. Material properties 138
8–7. Well operation schedule 147
C–1. Gauss points and weightings for 2-point rule in L8 elements 264
C–2. Gauss points and weightings for 8-point rule in L6 elements 265
C–3. Gauss points and weightings for 7-point rule in a linear triangle 265
I–1. Key numbers and words for user-specified units 414
I–2. Key words for debugging options 418
I–3. Key words for initialization of variables 421

LIST OF FIGURES

Figure Page

2-1. Fabric of granular assemblies (adapted from Dusseault and Morgenstern, 1978) 9

2-2. Athabasca oil sand peak and residual strengths illustrating curved Mohr-Coulomb envelope behaviour (adapted from Dusseault and Morgenstern, 1978) 10

2-3. Comparison of behaviour of Athabasca and Cold Lake oil sands under confined drained compression tests (adapted from Kosar, Scott and Morgenstern, 1987) 11

2-4. Schematic illustration of elastic, plastic collapse and plastic expansive strain components (for two yield surfaces) in drained triaxial compression test (adapted from Lade and Nelson, 1984) 13

2-5. Yield and plastic potential surfaces for plastic expansive increments (adapted from Lade and Nelson, 1984) 14

2-6. Three-dimensional representation of Mohr-Coulomb and Drucker-Prager yield surfaces in principal stress space (adapted from Owen and Hinton, 1982) 15

2-7. Two-dimensional π-plane (space diagonal normal to page) representation of Mohr-Coulomb and Drucker-Prager yield surfaces (adapted from Owen and Hinton, 1982) 16

5-1. Finite elements used by FESPS 48

5-2. Linear finite elements of time 63

5-3. Linear finite elements of time 63

6-1. Permeability ratio as a function of volumetric strain according to Kozeny-Poiseuille model 86

6-2. Forces acting on an arbitrary body 88

6-3. Coupling models at level of Newton-Raphson iterations 98

8-1. $1 \times 1 \times 8$ and $1 \times 1 \times 4$ grids used to investigate the effect of grid refinement on the linear brick element, held in an isotropic confining stress field. 134

8-2. Horizontal and vertical stress gradients displaying opposing magnitudes at shallower and greater depths. 139

8–3. View of simulation domain indicated heated and pressurized zone ($3 \times 3 \times 3$ discretization). . . . 142

8–4. View of heated zone and the immediately surrounding elements ($3 \times 3 \times 3$ discretization). . . 143

8–5. View of simulation domain indicated heated and pressurized zone ($5 \times 5 \times 3$ discretization). . . . 145

8–6. View of heated zone and the immediately surrounding elements ($5 \times 5 \times 3$ discretization). . . 146

8–8. View of $2 \times 2 \times 2$ Gauss quadrature points inside 8-noded brick elements. . . . 148

8–9. Profile of pressure (average block values) with simulation time at selected nodes near wellbore ($3 \times 3 \times 3$ fluid domain). . . . 149

8–10. Profile of temperature (average block values) with simulation time at selected nodes near wellbore ($3 \times 3 \times 3$ fluid domain). . . . 149

8–11. Growth of effective plastic strain with simulation time at selected Gauss points near the wellbore ($3 \times 3 \times 3$ fluid domain). . . . 151

8–12. Growth of volumetric strain with simulation time at selected Gauss points near the wellbore ($3 \times 3 \times 3$ fluid domain). . . . 151

8–13. Change in stress ratio, K_o, with simulation time at selected Gauss points near the wellbore ($3 \times 3 \times 3$ fluid domain). . . . 152

8–14. Change in vertical stress with simulation time at selected Gauss points near the wellbore ($3 \times 3 \times 3$ fluid domain). . . . 152

8–15. Change in Lode angle, θ, with simulation time at selected Gauss points near the wellbore ($3 \times 3 \times 3$ fluid domain). . . . 154

8–16. Change in least principal stress, σ_3, with simulation time at selected nodes near the wellbore ($3 \times 3 \times 3$ fluid domain). . . . 154

8–17. Change in the principal stress number oriented closest to the z-axis at selected nodes near the wellbore ($3 \times 3 \times 3$ fluid domain). . . . 156

8–18. Change in the angle of the principal stress oriented closest to the z-axis at selected nodes near the wellbore ($3 \times 3 \times 3$ fluid domain). . . . 156

8–19. Change in porosity, ϕ, with simulation time at selected nodes near the wellbore ($3 \times 3 \times 3$ fluid domain). . . . 157

8–19. Profile of pressure (average block values) with simulation time at selected nodes near wellbore ($5 \times 5 \times 3$ fluid domain; high rate). 159

8–20. Profile of temperature (average block values) with simulation time at selected nodes near wellbore ($5 \times 5 \times 3$ fluid domain; high rate). 159

8–21. Growth of effective plastic strain with simulation time at selected Gauss points near the wellbore ($5 \times 5 \times 3$ fluid domain; high rate). 160

8–22. Growth of volumetric strain with simulation time at selected Gauss points near the wellbore ($5 \times 5 \times 3$ fluid domain; high rate). 160

8–23. Change in stress ratio, K_o, with simulation time at selected Gauss points near the wellbore ($5 \times 5 \times 3$ fluid domain; high rate). 162

8–24. Change in vertical stress with simulation time at selected Gauss points near the wellbore ($5 \times 5 \times 3$ fluid domain; high rate). 162

8–25. Change in Lode angle, θ, with simulation time at selected Gauss points near the wellbore ($5 \times 5 \times 3$ fluid domain; high rate). 163

8–26. Change in least principal stress, σ_3, with simulation time at selected nodes near the wellbore ($5 \times 5 \times 3$ fluid domain; high rate). 163

8–27. Change in the principal stress number oriented closest to the z-axis at selected nodes near the wellbore ($5 \times 5 \times 3$ fluid domain; high rate). 164

8–28. Change in the angle of the principal stress oriented closest to the z-axis at selected nodes near the wellbore ($5 \times 5 \times 3$ fluid domain; high rate). 164

8–29. Change in porosity, ϕ, with simulation time at selected nodes near the wellbore ($5 \times 5 \times 3$ fluid domain; high rate). 165

8–30. Profile of pressure (average block values) with simulation time at selected nodes near wellbore ($5 \times 5 \times 3$ fluid domain; low rate). 168

8–31. Profile of temperature (average block values) with simulation time at selected nodes near wellbore ($5 \times 5 \times 3$ fluid domain; low rate). 168

8–32. Growth of effective plastic strain with simulation time at selected Gauss points near the wellbore ($5 \times 5 \times 3$ fluid domain; low rate). 169

8–33. Growth of volumetric strain with simulation time at selected Gauss points near the wellbore (5 × 5 × 3 fluid domain; low rate). 169

8–34. Change in stress ratio, K_o, with simulation time at selected Gauss points near the wellbore (5 × 5 × 3 fluid domain; low rate). 170

8–35. Change in vertical stress with simulation time at selected Gauss points near the wellbore (5 × 5 × 3 fluid domain; low rate). 170

8–36. Change in Lode angle, θ, with simulation time at selected Gauss points near the wellbore (5 × 5 × 3 fluid domain; low rate). 172

8–37. Change in least principal stress, σ_3, with simulation time at selected nodes near the wellbore (5 × 5 × 3 fluid domain; low rate). 172

8–38. Change in the principal stress number oriented closest to the z-axis at selected nodes near the wellbore (5 × 5 × 3 fluid domain; low rate). 173

8–39. Change in the angle of the principal stress oriented closest to the z-axis at selected nodes near the wellbore (5 × 5 × 3 fluid domain; low rate). 173

8–40. Change in porosity, ϕ, with simulation time at selected nodes near the wellbore (5 × 5 × 3 fluid domain; low rate). 174

8–41. Profile of pressure (average block values) with simulation time at selected nodes near wellbore (3 × 3 × 3 fluid domain; high rate). 176

8–42. Profile of temperature (average block values) with simulation time at selected nodes near wellbore (3 × 3 × 3 fluid domain; high rate). 176

8–43. Growth of effective plastic strain with simulation time at selected Gauss points near the wellbore (3x3x3 fluid domain; high rate). 177

8–44. Growth of volumetric strain with simulation time at selected Gauss points near the wellbore (3 × 3 × 3 fluid domain; high rate). 177

8–45. Change in stress ratio, K_o, with simulation time at selected Gauss points near the wellbore (3 × 3 × 3 fluid domain; high rate). 179

8–46. Change in vertical stress with simulation time at selected Gauss points near the wellbore (3 × 3 × 3 fluid domain; high rate). 179

8–47. Change in Lode angle, θ, with simulation time at selected Gauss points near the wellbore (3 × 3 × 3 fluid domain; high rate). 180

8–48. Change in least principal stress, σ_3, with simulation time at selected nodes near the wellbore ($3 \times 3 \times 3$ fluid domain; high rate). 180

8–49. Change in the principal stress number oriented closest to the z-axis at selected nodes near the wellbore ($3 \times 3 \times 3$ fluid domain; high rate). 181

8–50. Change in the angle of the principal stress oriented closest to the z-axis at selected nodes near the wellbore ($3 \times 3 \times 3$ fluid domain; high rate). 181

8–51. Change in porosity, ϕ, with simulation time at selected nodes near the wellbore ($3 \times 3 \times 3$ fluid domain; high rate). 183

8–52. Change in permeability, k, with simulation time at selected nodes near the wellbore ($3 \times 3 \times 3$ fluid domain; high rate). 183

8–53. Profile of pressure (average block values) with simulation time at selected nodes near wellbore ($3 \times 3 \times 3$ fluid domain; low rate). 185

8–54. Profile of temperature (average block values) with simulation time at selected nodes near wellbore ($3 \times 3 \times 3$ fluid domain; low rate). 185

8–55. Growth of volumetric strain with simulation time at selected Gauss points near the wellbore ($3 \times 3 \times 3$ fluid domain; low rate). 186

8–56. Change in stress ratio, K_o, with simulation time at selected Gauss points near the wellbore ($3 \times 3 \times 3$ fluid domain; low rate). 187

8–57. Change in vertical stress with simulation time at selected Gauss points near the wellbore ($3 \times 3 \times 3$ fluid domain; low rate). 187

8–58. Change in Lode angle, θ, with simulation time at selected Gauss points near the wellbore ($3 \times 3 \times 3$ fluid domain; low rate). 188

8–59. Change in least principal stress, σ_3, with simulation time at selected nodes near the wellbore ($3 \times 3 \times 3$ fluid domain; low rate). 188

8–60. Change in the principal stress number oriented closest to the z-axis at selected nodes near the wellbore ($3 \times 3 \times 3$ fluid domain; low rate). 189

8–61. Change in the angle of the principal stress oriented closest to the z-axis at selected nodes near the wellbore ($3 \times 3 \times 3$ fluid domain; low rate). 189

8–62. Change in porosity, ϕ, with simulation time at selected nodes near the wellbore ($3 \times 3 \times 3$

fluid domain; low rate). 191

8–63. Change in permeability, k, with simulation time at selected nodes near the wellbore ($3 \times 3 \times 3$ fluid domain; low rate). 191

8–68. Profile of oil-steam ratio during simulations. 193

8–64. Profile of cumulative steam injection during simulations. 194

8–65. Profile of oil production rates during simulations. 194

8–66. Profile of water production rate during simulations. 195

8–67. Profile of cumulative oil production during simulations. 195

8–69. Contours of effective plastic strain (%) in R553HR from basal layer of Gauss points in heated zone. 196

8–70. Contours of volumetric strain (%) in the high rate simulations from basal layer of Gauss points in heated zone. 197

8–71. Contours of vertical stress (MPa) in the high rate simulations from basal layer of Gauss points in heated zone. 198

B–1. Plot of residuals from the correlation of μ_{sc} with T 226

B–2. Plot of residuals from the correlation of k_{hsc} with T 227

B–3. Plot of residuals from the correlation of ρ_{sc} with T 228

B–4. Plot of residuals from the correlation of h_{sc} with T 229

B–5. Plot of residuals from the correlation of μ_{sv} with T 230

B–6. Plot of residuals from the correlation of k_{hsv} with T 231

B–7. Plot of residuals from the correlation of ρ_{sv} with T 232

B–8. Plot of residuals from the correlation of h_{sv} with T 233

B–9. Plot of residuals from the correlation of L_v with T. 234

B–10. Plot of residuals from the correlation of T with p 235

B–11. Plot of residuals from the correlation of p with T 236

B–12. Plot of residuals from Gottfried's correlation of μ_{sc} with T 237

B–13. Enthalpy of Athabasca bitumen as a function of temperature under saturated conditions . . 237

B–14. Solubility of methane in Athabasca bitumen as a function of temperature and pressure . . 238

B–15. Viscosity of methane as a function of temperature 239

B–16. Thermal conductivity of methane as a function of temperature 240

B–17. Enthalpy of methane as a function of temperature and pressure 241

B–18. Gas-liquid relative permeabilities 242

B–19. Oil-water relative permeabilities 243

B–20. Water-oil relative permeabilities 244

B–21. Oil-gas relative permeabilities 245

C–1. Example of a linear mapping between local and global coordinate systems. 251

C–2. Asymmetric weighting functions in a one dimensional element 253

C–3. Linear trial functions in one dimensional element 255

C–4. Weighting parameters in quadrilateral element 256

C–5. Weighting parameters in triangular element 257

C–6. L8 brick element with local coordinates and weighting directions 258

C–7. Linear trial functions for ξ-coordinate 259

C–8. The linear triangle 267

C–9. L6 triangular prism element with local isoparametric coordinate and weighting directions 269

D–1. Overview of FESPS program flow. 274

D–2. Overview of integration control subroutine. 277

D–3. Overview of calculation of shape functions. 279

D–4. Overview of calculation of shape function derivatives. 280

D–5. Overview of calculation of coordinate transformations. 281

D–6. Test geometry for bandwidth reduction example 302

F–1. Yield and plastic potential surfaces for plastic expansive increments (adapted from Lade and Nelson, 1984) 340

F–2. Local face tangent vectors on side of element 358

I–1. Outline of user input to run FESPS 409

I–2. Well definition in an L8 finite element 424

NOMENCLATURE

a	trial function coefficient
A, B, C, D	non-linear coefficient
A_s	surface area of grains [m^{-3}]
$\mathbf{B}$	matrix of space derivatives of trial functions
c	cohesion [Pa]
c_p	isobaric specific heat capacity [$\mathrm{J\ kg^{-1}\ K^{-1}}$]
c_ϕ	pore compressibility [Pa^{-1}]
$\mathbf{C}, C_{ijkl}$	stress-strain constitutive matrix [Pa]
$\mathfrak{C}, \mathfrak{D}, \mathfrak{E}, \mathfrak{L}, \mathfrak{M}$	linear differential operators
$\mathbf{D}, D_{ijkl}$	stress-strain constitutive matrix [Pa]
$d\varepsilon_{ij}^{e}$	increment of elastic strain
$d\varepsilon_{ij}^{p}$	increment of plastic strain
$d\bar{\varepsilon}^{p}$	increment of effective plastic strain
E	(Young's) modulus of elasticity [Pa]
f	constant terms in element equilibrium equation
F, G, H	arbitrary one-dimensional weighting functions
F_i	body force per unit volume, non-linear coefficient [$\mathrm{N\ m^{-3}}$]
$\bar{F}$	yield function
$\mathbf{g}$	acceleration due to gravity [$9.81\ \mathrm{m\ s^{-2}}$]
h	specific enthalpy [$\mathrm{kJ\ kg^{-1}}$]
G	Lamé constant (shear modulus)
$\bar{G}$	plastic potential
I_1, I_2, I_3	invariants of stress
$\mathbf{J}$	Jacobian of coordinate transformation
J_1, J_2, J_3	invariants of deviatoric stress
$\mathbf{k}$	absolute permeability tensor [m^2]
k^*	material hardening parameter
$\mathbf{K}$	stiffness matrix
k_h	thermal conductivity [$\mathrm{W\ m^{-1}\ K^{-1}}$]
K_o	three-dimensional stress ratio
k_Z	Kozeny constant (≈ 5)
$\mathbf{L}$	matrix of space derivative operators

L_i	trial functions in triangular elements
L_v	specific enthalpy of vaporization (latent heat) [kJ kg^{-1}]
M	molar mass [kg kmol^{-1}]
$\mathfrak{M}$	function value
$\mathbf{N}, N_i$	basis function
p	fluid (pore) pressure, steam saturation pressure [Pa]
P_c	capillary pressure [Pa]
p_i	initial pressure [Pa]
$\mathbf{q}$	heat flux [W m^{-2}]
q^*	source term [m^3 s^{-1}]
r	correlation coefficient, boundary term expression
R	residual, universal gas constant [8.314 kJ kmol^{-1} K^{-1}]
R_{so}	dissolved gas-oil ratio [m^3 m^{-3}]
s_{ij}	deviatoric stress [Pa]
S	liquid saturation [fraction PV], surface area [m^3]
S_{ij}	prescribed traction [Pa], non-linear coefficient
S_p	specific surface area per unit pore volume [m^3 m^{-3}]
t	time [s]
T	temperature [K], linear time function
T_i	prescribed tractions, [N m]
$\mathbf{u}$	displacement vector with components (u, v, w) [m]
u	pore pressure [Pa]
u, v, w	components of displacement vector [m]
U	specific internal energy [kJ kg^{-1}]
U_i	prescribed displacement
v_i	volume fraction
$\mathbf{v}$	intrinsic phase velocity [m s^{-1}]
V	volume [m^3]
V_b	volume [m^3]
V_r	volume [m^3]
V_p	volume [m^3]
$\mathbf{V}$	bulk relative (Darcy) velocity [m s^{-1}]
x, y, z	global coordinate directions
x	liquid mole fraction

$x_{i\alpha}$	mole fraction of phase α which is component i
w_1, w_2, w_3	triangular (area) coordinates
$\mathbf{w}$	velocity of phase interface [m s^{-1}]
W_i	weighting function
W_i^*	asymmetric weighting function
y	vapour mole fraction
z	gas deviation factor
α	coefficient of isobaric volumetric thermal expansion [K^{-1}], thermal diffusivity [m^2 s^{-1}]
α, β, γ	weighting function parameters
α'	Biot's constant
α_v	vapour density term [kg m^{-3}]
α_s	softening parameter
β	coefficient of isothermal compressibility [Pa^{-1}]
β_v	vapour density term [kg m^{-3}]
δ_{ij}	Kronecker delta
ϵ_{ij}	strain tensor
ϵ'_{ij}	deviatoric strain tensor
ϵ^0	initial strain
ϵ^0	initial strain
$\bar{\epsilon}_p$	effective plastic strain
ϵ_v	volumetric strain
Γ	surface boundary, Lode parameter
κ	volume averaged thermal conductivity [W m^{-1} K^{-1}]
λ	mobility ratio [(Pa.s)$^{-1}$], Lamé constant
μ	viscosity [Pa.s], shear modulus
ν	Poisson ratio
ω	additive quantity for volume averaging
Ω	integration domain (volume)
ϕ	porosity (fraction), arbitrary variable, angle of internal friction
ψ	arbitrary DOF, dilation angle
ρ	mass density [kg m^{-3}]
$\hat{\rho}$	mass density divided by $(1 + R_{so_r})$ [kg m^{-3}]
$\hat{\rho}$	molar density [kmol m^{-3}]
$\sigma_1, \sigma_2, \sigma_3$	principal stresses [Pa]

σ_m	mean stress [Pa]
σ_{ij}	total stress tensor [Pa]
σ'_{ij}	effective stress tensor [Pa]
σ'^0	initial effective stress [Pa]
θ	collocation point, Lode angle (stress invariant)
τ_{ij}	shear stress [Pa]
ξ, η, ζ	isoparametric (local) coordinates

SUBSCRIPTS

a	water
b	bitumen, bulk value
c	component, convective term
e	energy, elastic
f	fluid
g	gas
i, j, k	principal direction, phase
H, I, J, K	node number
ℓ, L	liquid phase (oil and water)
m	molar quantity
nc	non-convective term
o	oil
oct	octahedral (stress)
p	pore, plastic
r	rock matrix, reduced value
res	reservoir conditions
sc	steam condensate
sv	steam vapour
v	vapour
w	water
α, β, γ	phase, type of DOF, or equation index

SUPERSCRIPTS

e	(finite) element, elastic condition

n	time level
p	plastic condition
ν	iteration, time level

ABBREVIATIONS

bpd	barrels per day
cwe	cold water equivalent
DOF	degrees of freedom
EOS	equation of state
FDM	finite difference method
FEM	finite element method
FESPS	Finite Element Steam Parting Simulator
L6	6-noded element
L8	8-noded element
MWR	method of weighted residuals
PDE	partial differential equation
PV	pore volume
rb	reservoir barrels
SG	specific gravity
TBP	true boiling point

SYMBOLS

$\langle \cdots \rangle$	row vector
$\{ \vdots \}$	column vector
$[\vdots \;\; \vdots \;\; \vdots]$	two-dimensional array

Chapter I

INTRODUCTION

The recovery of heavy oil and bitumen from unconsolidated siliclastic sediments is considerable in central and eastern Alberta, western Saskatchewan, and elsewhere, particularly in the U.S.A. and Venezuela. While production has been established from these sediments for many decades, the degree of understanding of the reservoir mechanics giving rise to the production has varied considerably. In certain instances, the ability to predict a reasonably accurate production response to a given injection strategy has been poor. Lack of confidence in predictive ability, in turn, leads to the downgrading or outright cancellation of oil recovery projects, where an economic return might instead have been possible. This reflects a natural fear of making a large capital investment in what turns out to be an unprofitable venture.

One component missing from the conventional reservoir engineering analysis of these unconsolidated formations is the inclusion of geomechanical behaviour to a greater degree than usually considered in parameters such as pore compressibility. In the particular instance of thermal recovery of bitumen from the Athabasca oil sand by steam injection, the economic return depends mainly on the rate of mobilisation and subsequent recovery of bitumen in the sand. The rate of mobilisation of bitumen is clearly a function of the areal distribution of the injected heat energy—the wider the better, provided sufficient heat is available. The recovery of mobilised bitumen is thereafter primarily dependent on gravity drainage, hence on the combined vertical and horizontal conductivities of the porous medium in the zone of mobilised bitumen.

The injection of steam or another hot fluid into a formation saturated with essentially immobile bitumen can take place only through the parting or fracturing of the tar sand. This situation persists until the injected fluids reach a nearby well or a pressure sink such as an aquifer. The formation must be parted to inject steam in a classic cyclic steam stimulation process, unless communication to the nearby sink is established by some other means, e.g., through a layer containing mobile water. In each injection cycle one would hope that the parting was made in a relatively cool bitumen zone, where the heating would be most beneficial. The parting orientation and position is a function of several parameters—local ones such as lithology, pore pressure, local concentrations in total stress, material properties, and the far field effects of the regional principal stresses. The inclusion of all these effects should lead to a useful prediction of the distribution of heat and hence the rate of bitumen recovery. Usually, the variation in lithology is rarely known in detail, and when it is, it overwhelms our ability to include such fine detail. This information can only be incorporated on some averaged level. Pore pressures usually can be calculated with confidence, using the now classical tools of reservoir simulation. The incorporation of the local and regional stresses beyond that of pore compressibility remains only a recent development. Pore compressibility, used in traditional reservoir simulation, generalizes the matrix response to changes in effective stress by providing porosity change as a function of pore pressure.

The unconsolidated sands exhibit significantly different behaviour from that of typical cemented sandstone and limestones, which are characterised by elastic behaviour, linear and non-linear. The key differences are those of failure followed by irrecoverable strains and the phenomenon of dilation associated with shear deformation. These phenomena are the result of the interaction of individual grains grating against and riding up on top of each other to accommodate the applied loads. Both phenomena have significant consequences—the high possibility of a limiting load, and the disruption of the material fabric so that it is dilated and most likely has enhanced permeability. The dilation ultimately must increase the compressibility of the material because of the evolution of gases from solution. If the material is sheared then the pore compressibility will also change. Note that the appearance of even a 5% gas saturation in a shallow oil reservoir (p_i = 1.4 MPa) can cause an increase of more than 20-fold in the effective compressibility.

Chapter II

REVIEW OF LITERATURE

The topics covered in this research can be classified broadly as the application of finite elements to multiphase, thermal fluid flow, and the incorporation of geomechanics in traditional numerical thermal reservoir models, with particular emphasis on the behaviour of oil sands. Accordingly, the literature review has been divided into three sections: finite elements applied to multiphase flow, the material behaviour of oil sands, and the inclusion of geomechanics in current reservoir modelling technology. The section on oil sand material behaviour also constitutes a general description.

2.1 *Application of finite elements to multiphase flow problems*

The finite element method has been used successfully in solid mechanics and single-phase potential flow for many years. However, the method has been extended to multiphase flow only with differing degrees of success. Nevertheless, in the past decade, the conditions for successful finite-element simulation of multiphase flow in porous media have been coming to light. Two areas have been predominant—that of oil and water flow and that of steam and water flow. That is, in petroleum and geothermal reservoir simulation. A review of the most pertinent literature to the application of the finite element method to immiscible multiphase flow in porous media is made over the next few pages. The review is divided into two sections—isothermal and thermal flow.

2.1.1 *Literature survey for isothermal flow*

The first clear attempt to use the finite element method for fluid flow in porous media that appeared in the petroleum literature was in 1968, by Javandel and Witherspoon. They formed a variational principle for the flow of a single fluid through an isotropic, homogeneous porous medium. The numerical solutions were compared with the Theis (analytical exponential integral) solution for infinite, bounded and layered radial systems with either constant terminal rate or pressure constraints and were found to be good.

In 1969, Cavendish *et al.* applied the Galerkin principle to the two-dimensional elliptic partial differential equation describing the flow of a single fluid. Bilinear Hermitian functions were used to approximate the pressure distribution and pressure gradients in rectangular elements. Wells were introduced as point sources or sinks to form non-smooth subspaces. More accurate approximations to the pressures around wells were made by adding basis functions, such as an analytical radial solution, to the element. The workers concluded that their application of Galerkin's method was more accurate and computationally more efficient than the finite difference method.

Zienkiewicz and Parekh (1970) investigated the solution to transient field problems in two and three dimensions by isoparametric finite elements. The physical example given was the transient heat conduction problem with mixed boundary conditions. The workers applied the Galerkin principle to write the weighted residual equation which was then solved for four examples using linear and quadratic isoparametric elements in two and three dimensions. The examples compared well with the analytical solutions.

The first multiphase problem to be solved by the application of Galerkin's method was presented by McMichael and Thomas in 1973. The workers analyzed three-phase flow in a two-dimensional domain subdivided into linear finite elements. Reportedly, no difficulties were encountered in finding the solution at each time step. The evaluation of all the reservoir properties at each quadrature point for numerical integration appeared to obviate the need for upstream weighting for numerical stability. This result is not in accordance with later studies of the multiphase flow problem by the finite element method.

Huyakorn and Pinder (1977a) applied a weighted residual method to the two-phase flow problem in one and two dimensions. This formulation required asymmetric weighting functions which degenerated to the basis function when all nodes in an element were accorded equal influence. Otherwise, upstream nodes were given more weighting in the asymmetric scheme. Such an asymmetric scheme is known as the Petrov-Galerkin method. In addition, the influence of the accumulation terms at neighbouring nodes was removed by lumping the element mass matrix to give a more stable solution. A two-dimensional example of a five-spot was used to show that grid orientation effects were negligible in this solution scheme. In 1978 Huyakorn and Pinder further explained the asymmetric weighting functions, and the conditions for their derivatives to be valid. The results using the asymmetric weighting scheme were compared with other upstream weighting schemes for the finite element method. The Newton-Raphson and fully-implicit chord slope incremental methods were compared as means of linearizing the original equations. Both methods proved stable at large time step values.

The use of higher order approximation functions, rather than upstream weighting, was chosen by Spivak *et al.* (1977) for a two-dimensional, two-phase flow problem. The Galerkin procedure was successfully applied to the IMPES formulation of the governing equations in the presence of favourable and unfavourable mobility ratios. Numerical dispersion at the front was less in both cases than with finite differences. Subsequently, a semi-implicit linearization of the saturation equation was employed to improve solution stability. The approximation to the pressure solution was augmented by logarithmic functions in the vicinity of point sources and sinks. Grid orientation effects were not observed when the mesh for a quarter five-spot was rotated 45°. A three point (in one dimension) Gaussian quadrature scheme was used to evaluate the integral expressions. No improvement in the solution was observed when higher order quadrature schemes were used. In 1977 Hayes, *et al.*, also discussed the approximation to the pressure solution around a well. An improvement was obtained by locally adding a logarithmic distribution around the singularity.

Wilson and Casinder (1978) studied the problems arising from the application of Galerkin's method to the two-phase problem, when cubic Hermitian functions are used to approximate the primary variables. Wells were represented by point sources and sinks. The mass matrix was lumped for solution stability. The solution to the Buckley-Leverett problem could be improved either by using a constant pseudo-capillary pressure gradient or by adding upstream weighting to the quadrature points. A modified fractional flow curve was also required.

Lewis *et al.* (1978) applied the Galerkin method to the two-phase problem in two dimensions. The problem was reformulated in terms of the half-sum and half-difference of the wetting and non-wetting phase potentials. A predictor-corrector method was then used to solve the problem in the time domain. Solution surging, or oscillations, were noticed near the shock in the Buckley-Leverett example. Mesh refinement at the front reduced the surging, as did lumping the mass matrix. However, mass matrix lumping reduced the accuracy of the solution.

In 1979, Dalen presented a simplified finite element model for gas or water-oil flow, using linear triangular elements. The mass matrix was lumped for stability. A method for using the mobilities at upstream nodes in the element flow matrix was incorporated into the Galerkin scheme. The model was applied successfully to a two-dimensional example of a waterflood. In that case the grid orientation effect was considered to be comparatively small.

White *et al.* (1981) took the same approach as Lewis, *et al.*, (1978), but used an upstream Gauss quadrature scheme. This was achieved by bringing the mobilities into the integral and using the values at upstream Gauss points. The scheme was not successful in certain finite elements of the five-spot example. The reformulation of the problem in terms of half-sums and half-differences of the two phase potentials decoupled the primary variables. This permitted a large reduction in the amount of work required to solve the equation set.

Allen and Pinder (1983) used orthogonal collocation to solve the one-dimensional convective-dispersion transport equation. Upstream weighting of the collocation points was necessary to find the correct (analytical) solution in the Buckley-Leverett case. Upstream weighting was achieved by adding a quartic term to the Hermite cubics, which resulted in an asymmetric interpolation scheme.

Also in 1983, Mohsen and Pinder investigated one-dimensional collocation with finite elements in order to solve the Buckley-Leverett problem. Hermite cubic polynomials were used everywhere but at the front, where the variables were interpolated in one or two adaptive elements by Hermite quintics. The asymmetric interpolation scheme was used to weight the upstream collocation points.

Allen (1984a) discussed the validity of using upwinding (upstream weighting) in the finite element method. Upwinding was required when centered schemes failed to "impose proper uniqueness criteria." Cauchy problems such as the Buckley-Leverett problem might only possess weak solutions (solutions satisfying only the integral form of the defining equation, thereby admitting the possibility that the solution may not be sufficiently differentiable to satisfy the defining equation everywhere). In that case, the correct shock (front) requires the weakest solution to depend continuously and stably on the initial data.

Chavant *et al.* (1984) used mixed finite elements to investigate two-dimensional, two-phase incompressible flow. The flow equations were formulated in terms of a global pressure and a reduced saturation. The global pressure was approximated by a constant function, and the reduced saturation by a linear function. Upwinding of the solution equation was introduced at the upwind boundary in the form of a Godunov saturation. The simulated quarter five-spot examples gave good results.

Cohen (1985) compared several finite element methods by applying them to one- and two-dimensional surfactant/polymer simulators. He considered the Petrov-Galerkin, the standard upwind and the negative-dispersion methods. Linear and bilinear approximation functions were used, with analytical and 2×2 Gaussian integration. In all the examples the Petrov-Galerkin formulation provided sharper fronts. The standard upwind method (using upstream quadrature points) dispersed the front considerably. The addition of negative dispersion to the standard Galerkin formulation provided a solution between the other two methods. Straightforward application of Galerkin's method yielded a sharp front with oscillations in the downstream solution.

Gulbrandsen and Wille (1985) applied the Galerkin method directly to two-dimensional, two-phase flow. The Newton-Raphson method was used to linearize the weighted residual form, which was approximated in time by backward Euler differences. The spatial domain was divided into rectangles and approximated by bilinear functions. A sharper front was noticed when the capillary pressure was not simply a constant function of saturation, but oscillations in the solution still occurred downstream of the front. However, no serious solution instability occurred. An interesting example was the displacement of oil through a reservoir with a sinusoidally-shaped cross-section.

Ewing (1989) proposed mixed elements solving for pressure and velocity as being most appropriate for miscible and immiscible two-phase reservoir flow problems. Velocity was chosen as the primary variable to ensure that it remains a smooth function in space, despite step changes in reservoir properties governing flow. Higher order functions were considered useful if they were modified in the region of sources or sinks. Operator splitting is advocated to excise the purely hyperbolic portion of the equation, which can develop shock solutions. This portion is solved in time using the method of characteristics. The paper is a review of previous work done separately on each of the main sections.

2.1.2 *Literature survey for thermal flow*

Faust and Mercer (1976) investigated the solutions to the geothermal problem using finite difference and finite element methods. Water only and water/steam systems with zero capillary pressure were considered in an areal model. Phase change in a block or element was forced to occur at the end of a time step for greater solution stability. Oscillations in the finite element solution to the two-phase problem occurred. No upstream weighting scheme was employed in the two-dimensional, linear and quadratic, isoparametric quadrilateral finite elements. Newton-Raphson iteration was not successful when applied to the finite element method approximation to the two-phase problem.

Huyakorn and Pinder (1977b) introduced asymmetric weighting functions for convective terms to form a Petrov-Galerkin scheme for the two-phase geothermal problem. The asymmetric weighting functions provided extra influence from upstream nodes in linear triangles and quadrilaterals. The mass matrix was lumped to improve the solution stability. A novel scheme to evaluate the derivatives of non-linear coefficients at a node during a phase change was presented.

Aktan and Farouq Ali (1978) applied Galerkin's method to the flow of hot water in a two-dimensional model using linear triangles. The mass and energy balances (in p_w and T) were solved in a cyclic iteration to a given tolerance. Changes in the principal stresses in the formation were then evaluated using thermoelastic theory for plane strain. Gravitational and capillary forces were not included in the analysis. The study was aimed at the evaluation of stresses in the system.

Voss (1978) applied the mixed Galerkin/Petrov-Galerkin method to two-phase geothermal flow in one- to three-dimensional, linear, isoparametric elements. The governing equations were developed by volume averaging point conservation equations. The weighted residual form of the equations were then solved for incremental change in p_w and h. An inverse iterative scheme was provided for those nodal solutions crossing the phase boundaries which did not restrict the time step size.

Karahanoglu *et al.* (1985) developed a fully-coupled finite element solution for hot water flow in a two-dimensional deforming reservoir. The mixed Galerkin/Petrov-Galerkin method was used to define the weighted residual form. Asymmetric weighting functions were applied to the convective terms only. The time domain was discretized into linear finite elements.

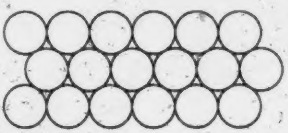
Lewis *et al.* (1985) used the Galerkin method to find the weighted residual forms of the water flow and energy equations in two dimensions. Bilinear elements were used to model hot water flooding for thermal oil recovery. Linear and higher order infinite elements modelled the heat losses from the reservoir in all directions. Artificial diffusion was introduced along streamlines to negate any grid orientation effects. The solutions were found efficiently at the end of each time step using an alternating direction solution algorithm.

2.2 *Description of the geomechanical behaviour of oil sands*

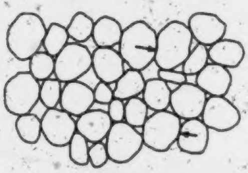
The geomechanical study of the oil sands of Alberta is effectively the study of the soil mechanics of a dense sand. This conclusion was not widely accepted until the completion of a series of research programs at the University of Alberta Civil Engineering department in the late 1970s and early 1980s under the direction of Drs. Morgenstern and Scott. In particular, the perception of bitumen as a cementing material was widely held until the last decade, as many geologists and petroleum engineers failed to recognise that their core samples had undergone expansion of the sand (soil) matrix due to pressure relief and gas evolution. It is now recognised that the oil sands must be considered a particulate material, for which the behaviour can be described by an appropriate stress-strain model. In particular, the description of the essence of oil sand behaviour as a c-ϕ material — through the Mohr-Coulomb parameters of cohesion c and angle of internal friction ϕ — is now common.

The behaviour of soils taken to failure has been studied for over a century, although the most notable advances have been made since the second world war. The study of soil mechanics is therefore pertinent to the behaviour of oil sands during bitumen recovery schemes, where loadings are changed because of the application of temperature increases and pore pressure changes. There are several good books on soil mechanics which are requisite reading for anybody interested in understanding how sands will react to these loads and how this might affect the bitumen recovery scheme (e.g., Wood, 1990). Some of the basic geomechanical concepts are brought out in this section.

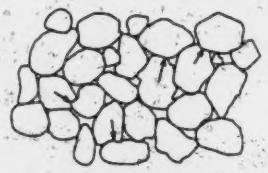
Athabasca and Cold Lake oil sand geomechanical behaviour has been the object of significant



(a) Hexagonal close-packed spheres.
Point contacts.



(b) Densely packed rounded sand.
Point contacts, with some straight contacts (arrows)



(c) Athabasca oil sand
Point contacts, with many straight and
interpenetrative contacts (arrows)

Figure 2–1

Fabric of granular assemblies (adapted from Dusseault and Morgenstern, 1978)

study since the mid-1970s. Dusseault (1977) showed that the Athabasca oil sands had an extremely stiff structure (i.e., a large modulus of elasticity) in the undisturbed state, accompanied by a large degree of dilation when loaded to failure and subsequent yield. Agar (1984) examined the stress-strain behaviour for different stress paths and at elevated pressures and temperatures. Kosar (1989) continued this work and noted some essential differences in the geomechanical behaviour between the oil sands of Athabasca and Cold Lake. These differences were discussed by Kosar *et al.* (1987).

In the case of the Athabasca oil sand the investigators noted the very high initial elastic (Young's) modulus of the confined and undisturbed material. This was attributed to its extreme compactness providing extensive grain-to-grain contact so that the stiffness of the sand skeleton is close to that of the grains themselves. This grain orientation is compared to ideal and rounded sand grains in Figure 2–1. The angularity of the Athabasca sand grains also illustrates why significant

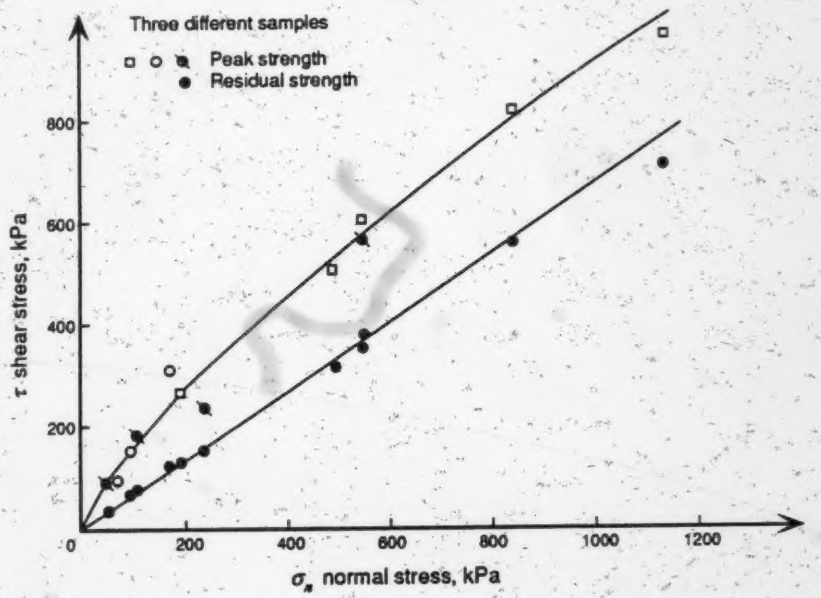


Figure 2–2

Athabasca oil sand peak and residual strengths illustrating curved Mohr-Coulomb envelope behaviour (adapted from Dusseault and Morgenstern, 1978)

dilation can be expected as the sand is sheared. The results of a typical direct shear box test by Dusseault are presented in Figure 2–2. The Mohr-Coulomb failure envelope on this shear-stress vs. normal-stress diagram is seen to be curved, indicating different mobilized strengths at failure, and possibly different modes of failure, under different levels of effective stress. The friction angle, an indicator of the slope of the Mohr-Coulomb failure envelope, varies considerably as the effective confining stress is increased. The residual (post-failure) friction angle for Athabasca oil sand was considerably less than the initial values. The extrapolation of the peak strength curve to the shear stress axis gives the impression of an apparent cohesion, the idea of which is only dispelled by data points taken at low values of normal stress.

In contrast, the Cold Lake sand was less stiff, and underwent little loss of strength after the initial yield. The dilatant behaviour exhibited by the two oil sands also differed. Cold Lake oil sand did not exhibit the dilatant behaviour of the Athabasca oil sand; instead, it displayed contractile

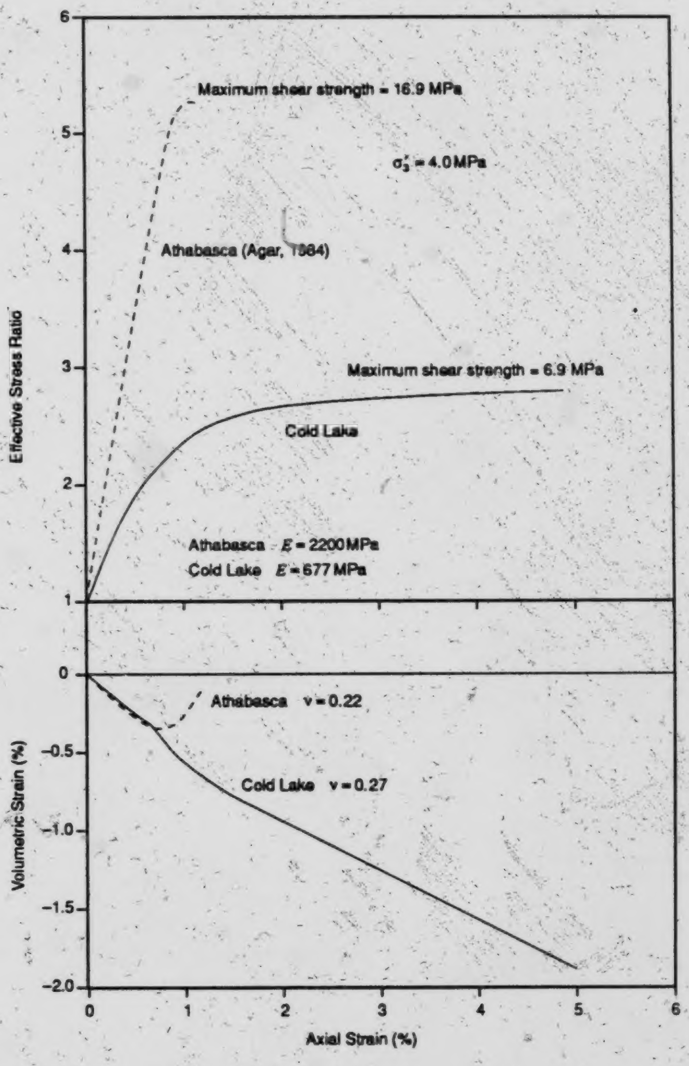


Figure 2–3

Comparison of behaviour of Athabasca and Cold Lake oil sands under confined drained compression tests (adapted from Kosar, Scott and Morgenstern, 1987)

behaviour throughout the triaxial confining stress path. As a result, the change in pore pressure during undrained testing of Cold Lake sand remained fairly constant as the degree of axial strain was increased. In the case of Athabasca oil sand, the increase in volume was apparent due to the sharp decrease in pore pressure. The net confining stress at failure is thus affected in different ways for each of the two oil sands. Figure 3 illustrates the differences in geomechanical behaviour of the two types of sand. The differences in behaviour are explained in terms of the different mineralogy. The Cold Lake deposit has a greater proportion of weaker minerals than the highly siliceous Athabasca sand. These weaker minerals are prone to crushing at the levels of deviatoric stress considered by the investigators.

At ambient temperatures of the in-situ oil sands, no measurable cohesion was observed (Dusseault, 1977). However, at high stresses the altered Mohr-Coulomb failure envelope gives the impression that positive cohesion exists (see Figure 2). This is only an apparent cohesion because, as noted, failure at lower stresses clearly indicated that there was no cohesive strength. Agar (1984) did note the development of low levels of cohesion in baked samples of Athabasca oil sand. Conversely, the increase in susceptibility to failure of individual grains due to an increase in temperature was marked for Cold Lake oil sand.

The main difficulty in describing the behaviour of the oil sand over a wide range of stresses lies in the post-failure behaviour where irrecoverable strains occur, and the development of dilation or positive volumetric strains. Dusseault and Morgenstern (1978) showed that the Mohr-Coulomb failure envelope is not a straight line through the origin, which would be typical of a particulate material with only intergranular failure where the grains slide against each other with a measurable friction. A failure envelope which is curved indicates that an additional failure mechanism is occurring—intragranular failure where individual grains are sheared. This is common behaviour of systems where grain breaking occurs at the higher levels of stress (Vesic and Clough, 1968, and Feda, 1982). This is consistent with the observations of Dusseault, where the (actually cohesionless) material has an apparent cohesion intercept at higher normal stresses. This behaviour could be reproduced mathematically by the expedient of making the cohesion and the friction angle a function of normal stress. The failure envelope then represents that of an altered material, raising the question of what the envelope should be for a system unloaded subsequent to failure. More sophisticated methods also exist as part of a larger constitutive modelling approach.

Modelling the oil sand behaviour must include the two significant features other than failure — irrecoverable strains and dilatancy. A suitable model must account for the deformation history of the material, particularly if the stresses are to be cycled through loading and unloading. The elastoplastic formulation incorporates all these features naturally. The relation between stress and strain is modified using a yield surface, essentially a function defined by the material properties and

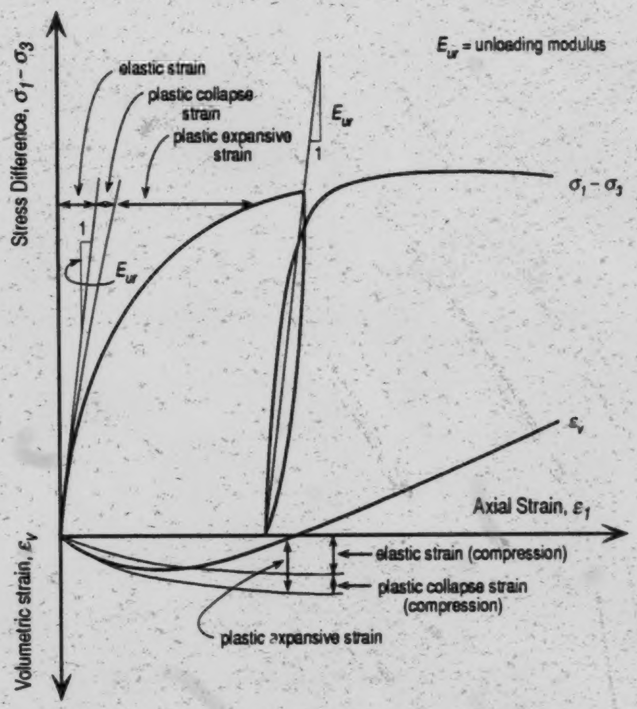

E_{ur} = unloading modulus
Stress Difference, $\sigma_1 - \sigma_3$
elastic strain
plastic collapse strain
plastic expansive strain
E_{ur}
1
$\sigma_1 - \sigma_3$
1
E_{ur}
ε_v
Axial Strain, ε_1
Volumetric strain, ε_v
elastic strain (compression)
plastic collapse strain (compression)
plastic expansive strain

Figure 2–4

Schematic illustration of elastic, plastic collapse and plastic expansive strain components (for two yield surfaces) in drained triaxial compression test (adapted from Lade and Nelson, 1984)

the stress and strain history.

Most previous work in this field has used variants of elasticity theory to model the stress-strain relationship of oil sands (see, for example, Agar *et al.*, 1987, Vaziri, 1988, and Settari *et al.*, 1989). Increasing the order of elasticity—hyperelasticity (making it non-linear)—cannot, alone, describe the effect of shear stress on the volumetric strain (dilation) or the effect of the stress path (Feda, 1982). The elastic stress-strain relation is improved, however, by writing it in the incremental form of the hypoelastic constitutive equations

$$d\boldsymbol{\sigma} = \mathbf{A}\,d\boldsymbol{\varepsilon} \tag{2.1}$$

where $d\boldsymbol{\sigma}$ and $d\boldsymbol{\epsilon}$ are respectively the vectors of increments of stress and strain, and **A** is the hypoelastic material behaviour matrix. This model works for a monotonically increasing load but cannot predict the irrecoverable strains observed on unloading. It does, however, describe well the elastic portion of the stress-strain behaviour. The hypoelastic model can be augmented with a relationship for volumetric strain as a function of axial strain. Previous research has included this model (Agar *et al.*, 1987) but was unsuccessful, in that form, at describing the dilatant volume changes observed after about 0.5% axial strain for Athabasca oil sand. A more complete argument for the choice for oil sands of an elastoplastic model over a hypoelastic model with extensions for dilation can be found in Wan *et al.* (1989).

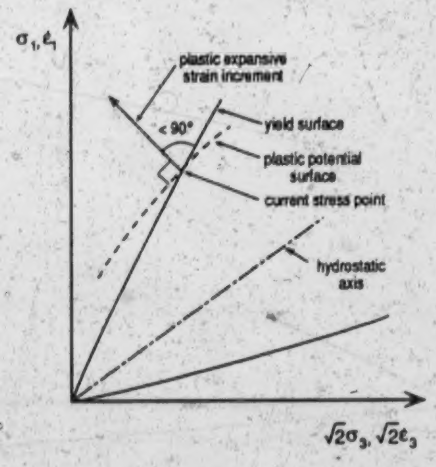

σ_1, ϵ_1
plastic expansive strain increment
< 90°
yield surface
plastic potential surface
current stress point
hydrostatic axis
$\sqrt{2}\sigma_3, \sqrt{2}\epsilon_3$

Figure 2–5

Yield and plastic potential surfaces for plastic expansive increments (adapted from Lade and Nelson, 1984)

The appropriate model for a complete description of oil sand behaviour is the elastoplastic one, combining elasticity with incremental plasticity. This model will generate the appropriate plastic strains and the accompanying dilation. An idealized response of a sand to a drained triaxial compression test (Lade and Nelson, 1984) is shown in Figure 2–4. The total strain increment, $d\varepsilon_{ij}$, is made of an elastic component, $d\varepsilon^e_{ij}$, and a plastic component, $d\varepsilon^p_{ij}$.

$$d\varepsilon_{ij} = d\varepsilon^e_{ij} + d\varepsilon^p_{ij} \tag{2.2}$$

It can be seen in Figure 2–4 that the unloading path results only in the recovery of elastic strains, while the plastic strains, by definition, remain irrecoverable. Increased straining also causes the dilation evident in the plot of volumetric strain vs. axial strain.

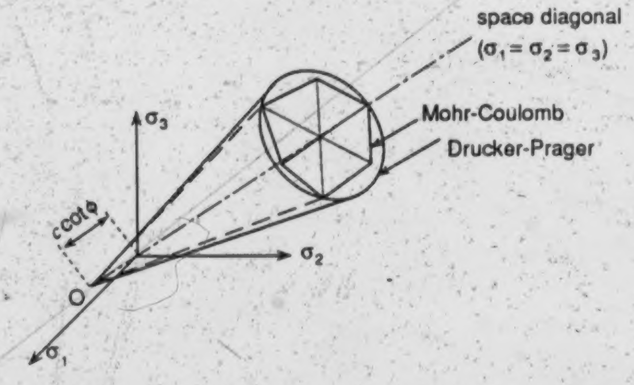


Figure 2–6

Three-dimensional representation of Mohr-Coulomb and Drucker-Prager yield surfaces in principal stress space (adapted from Owen and Hinton, 1982)

The initial yield surface is defined in terms of the material properties and the stress and strain history. Once the stress locus arrives at the yield surface it may not pass through it, and plastic straining is initiated. The subsequent yield surface may further change as a result of the plastic straining, expanding in a manner defined by the hardening law. The strain directions are given as normal to a second surface called the plastic potential, as shown in Figure 2–5. The simplest situation is where the yield and plastic surfaces coincide and the increment of strain is thus also normal to the yield surface—termed associated plasticity. The direction of the plastic strains controls the values of the volumetric strain, and the assumption of associated plasticity can lead to too great a predicted dilation. In that case, a plastic potential surface can be defined apart from the yield surface—so-called non-associated plasticity—which will provide volumetric strains in closer agreement with the experimental evidence. The plastic potential surface still passes through the same point in the stress space where the yielding is occurring (on the yield surface). Usually the plastic potential surface is described by the same function as the yield surface but with one or more parameters taking modified values to fit the experimental results.

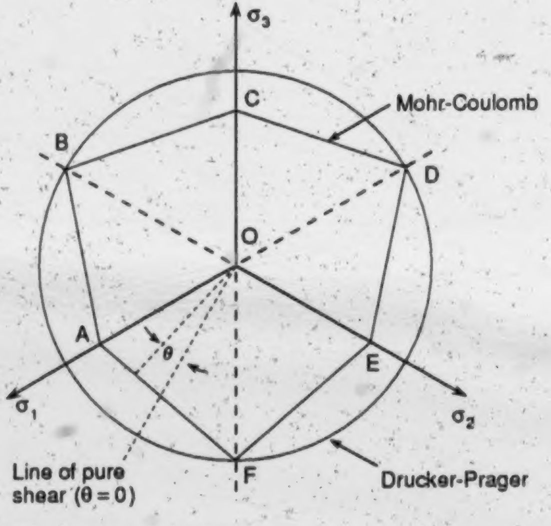


Figure 2–7

Two-dimensional π-plane (space diagonal normal to page) representation of Mohr-Coulomb and Drucker-Prager yield surfaces (adapted from Owen and Hinton, 1982)

The yield surface may remain static, coincident with the initial yield surface, or it may move away from it. The latter behaviour is known as hardening. Simple hardening models in use today are confined to an isotropic expansion of the yield surface, and a translation of the entire yield surface known as kinematic hardening.

The Mohr-Coulomb yield surface in three-dimensional space is illustrated in Figures 2–6 and 2–7. This is a preferred model for soils and rocks, exhibiting different yield strengths in compression and tension. The Drucker-Prager yield surface also shown in Figures 2–6 and 2–7 is an approximation to the Mohr-Coulomb surface in one of the two failure modes—tension or compression. Both models feature isotropic hardening with higher stresses, as can be seen in the widening neck of the yield surfaces.

2.3 *The inclusion of geomechanical behaviour in reservoir models*

Geertsma (1957) combined the previous work of Biot (1941) and Gassman (1951) to develop the equations of poroelasticity in a more straightforward manner. He clearly defines and relates the rock bulk and pore compressibilities, and describes the boundary conditions and procedure to determine the correct parameters defining the compressibilities. The procedures are divided into different sets of parameters for the three lithological types of sandstone, shale, and limestone. The paper is notable for the clarity of the presentation of poroelasticity as it is applied to petroleum engineering.

Geertsma (1966) reviewed the applications of poroelasticity in petroleum engineering. An analogy is presented between poroelastic and thermoelastic theory, to take advantage of the many solutions under different boundary conditions that have already been published for the thermoelastic theory. The concept of the nucleus of strain for volume elements is described. This concept is applied to the phenomenon of reservoir compaction and subsidence to calculate surface displacements for the examples of pore pressure depletion (leading to subsidence) and to reservoir heating (leading to uplift). Geertsma also discusses borehole stress and pressure solutions, and the different rock mechanics problems of hydraulic fracturing in impermeable and permeable formations. In hydraulic fracturing he analyses the degree of rock expansion in a permeable formation due to leak off using the nucleus of strain theory, and concludes that it is not significant.

Raghavan (1972) derived the one-dimensional consolidation expression coupled with fluid flow, and compares it to the earlier work of the geomechanical scientist Terzhaghi. The general solution is obtained from the partial differential equations describing the flow of fluid and material displacement using a transform to convert it to an ordinary differential equation. Raghavan has prepared a significant review of literature to that time, and stresses the importance on the final solution of keeping the mass flux ($\mathbf{v}_r$) in the formulation. The influence of pressure sensitive rock properties is shown on previously published well test analyses.

Finol and Farouq Ali (1975) presented a two-phase, two-dimensional flow model using finite differences which included the effects of compaction on fluid flow and the prediction of subsidence. The problem was formulated in two discretized equations for oil and water flow, and one analytical equation for poroelasticity which was numerically integrated. The variation of permeability as well as porosity was considered in the analyses of the effect of compaction on ultimate recoveries. The investigators concluded that ultimate recoveries of oil increased with compaction.

Closmann and Phocas (1978) showed that, given time to heat enough rock mass, a principal stress regime preferring vertical fractures ($\sigma_v = \sigma_1$, i.e., in a compression-positive stress regime, the least principal stress is the lesser horizontal stress) could be rotated so that horizontal fractures would be formed preferentially. The heating caused volumetric thermal expansions. These expansions could be accommodated vertically by translation of the overlying free surface, but set up horizontal loads through the elastic reaction, thus causing greater horizontal stresses. No pore pressure distribution was included in the analysis, but the paper clearly showed the ultimate effect of heat transfer from the fracture to the surrounding rock.

Rattia (1980) studied the effect of compaction in a three-dimensional steam simulator. Compaction was included by replacing the pore compressibilities by the terms described by Geertsma (1957), which include a compaction coefficient. By varying the value of the compaction coefficient, he was able to conclude that an increase in oil recovery accompanied an increase in compaction coefficient, and that compaction enhanced cyclic steam stimulation more than it did steam flooding.

The cause and significance of principal stress rotation, such as the one described by Closmann and Phocas, was further investigated by Dusseault and Simmons (1982). In a numerical study of McMurray Formation sands the authors showed that an appropriate distribution of pore pressure gradients around a fracture could alter locally the direction of the minimum principal stress. In this case, the leak-off of fluid constitutes a loading on the material with effects similar to the previously described thermoelastic straining.

Ito (1984) appears to have been the first to recognize and implement the dispersed and widespread nature of matrix deformation in the oil sands in numerical reservoir simulation. He had no predictive model to help him quantify the effects, but was able to choose appropriate parameters by history matching field data. Ito varied porosity between the initial value and a maximum — described by a multiplication factor (1.04 in the examples given) — as a function of pressure, between a given pair of pressures. He also increased the mobility of the injected fluid by two orders of magnitude during the first injection period, and by 20 at other times. This allowed him to model microchannels, as he termed them, dispersed high-permeability streaks created by the deformation of the oil sand. It also enabled him to avoid large changes in porosity that were required in contemporary simulation studies to permit the injection volumes of steam observed in the field. Lastly, a super-upstream weighting factor was employed to disperse energy further, as would be expected from the presence of microchannels, while ensuring that the finite difference blocks did not respond with an increase in temperature as quickly as they would without the super-upstream weighting. While Ito could history match field data, he had no way of defining the deformed region or the number and size of the microchannels.

Vaziri (1988) coupled thermal single-phase flow to a non-linear elastic model, with the solution formulated in two-dimensional finite elements by combining the force equilibrium equations with Darcy's law. He was able to model the effect of a second, gas, phase by including its compressibility in the definition of the bulk modulus (inverse compressibility). The quantity of gas as a separate phase was found using a Henry's law coefficient. As an example, Vaziri examined the case of the response near a wellbore to repeated injection, soak, and production cycles, predicting stresses, radial displacements and pore pressures radially away from the wellbore.

Schrefler and Simoni (1988) investigated the flow of water in a partially unsaturated soil by considering the gas phase (air) to be at constant pressure. Gas solubility was included, but capillary pressure was neglected. The authors used a Crank-Nicholson discretization of the time derivatives and quadratic isoparametric finite elements to solve the example of a one-dimensional outflow from a consolidating column.

Settari *et al.* (1989) described a new model to quantify leak-off rates from fracture faces in oil sands. The model incorporates a failure criterion from which the amount of dilation is calculated. The isotropic bulk compressibility is found from a non-linear empirical curve as a function of effective confining stress (effective stress). The elastic properties (Young's modulus and Poisson's ratio) are modified as a function of stress and failure. As part of the model implementation, various parametric analyses were performed using a linear elastic fracture model (LEFM) simulator. It was noted that the fracture length in the LEFM models was strongly dependent on fracture leak-off rates — a result to be expected because this factor mostly controls the pressure distribution in the fracture away from the wellbore. Thus the thrust of the work was to improve the analysis of leak-off from the fracture face using material failure. Stress and flow are solved sequentially, such that the stress solution lags behind one time step in a staggered formulation. The stress and fluid flow models are coupled via the bulk compressibility term, c_b. This term is kept updated at the $n+1$ time level by making it a function of minimum effective stress, $\sigma'_{min} = \min(\sigma_x, \sigma_y, \sigma_y)^n - \alpha' p^{n+1}$. Failure was applied as an instantaneous one-time increment of deformation, arbitrarily assigned a value of 1% or 2%, along with an arbitrary constant increase in permeability. The staggered scheme is claimed to be accurate and convergent. The authors acknowledge the plastic nature of the real deformations, and present their analysis only as an approximation. The new leak-off model is used to generate leak-off rates which are parametrized using a hydraulic diffusivity term 50m away from the well, which is in turn used to govern leak-off in the LEFM simulator. Only two-phase isothermal flow with a one-dimensional stress analysis is considered here.

Settari (1988), referring to the work of Settari *et al.* (1989) which was originally presented earlier in 1988, described some of the features of leak-off into oil sands from a fracture face. High leak-off rates which are non-linear due to oil sand deformation is given as the cause of shorter fracture lengths than could be predicted using conventional LEFM simulators. Behaviour at the fracture tip was modified to account for the high viscosity of the bitumen, which dissipates the stress concentrations typically found there, and the impact of loose sand carried from the failed zone on the fracture face to the fracture tip where it packs off like a proppant bank. Sensitivity tests on history matches of reported minifrac tests investigated modifications increasing the effective permeability for leak-off by one order of magnitude, increasing the system compressibility by a factor of two, removing all the tip effects, and increasing the formation stiffness (E) by a factor of two. These changes all affected the fracture closure time.

Dusseault and Rothenburg (1988) reviewed the effect of thermal loading and pore pressure changes around a wellbore on dilation and permeability. The authors describe the physical process of deformation in terms of particulate media, noting earlier work (Mori and Tamura, 1986) on permeability changes in slightly permeable systems (sand grouted with cement). They draw conclusions that effective water permeability would increase one or two orders of magnitude with dilation as the thickness of the water film coating the grains would increase by a factor of two. While the resultant change in effective permeability is likely, the analysis overlooks the fact that bitumen is more compressible than water and that gas is more likely to fill the pore volume newly created by the dilation. The authors continue to document the changes likely from shear failure, including the localization of shear, and the growth of the shear zone from the edge of a hydraulic fracture due to the altered stress state and the increased pore pressures.

Settari (1989) extends the modelling efforts of Settari (1988) and Settari *et al.* (1989) to thermal flow. The approach is the same: material properties are varied non-linearly as a function of minimum effective stress, and the solid and fluid models are coupled through the bulk compressibility. The staggered formulation is convergent and accurate. A one-dimensional analysis of leak-off from a fracture is presented and compared to a similar analysis using linear elastic properties. The non-linear response is shown to give a completely different pressure distribution, and is described as the effect of shear failure of the oil sand.

Xikui-Li (1990) *et al.* studied the interaction of two-phase immiscible flow with the porous medium. The model is based on the mass conservation equations as a function of intrinsic phase velocities and the solid and liquid equilibrium equations for a unit element of porous medium. The equations are discretized using finite elements, with a quadratic scheme for the time domain. The derived linear system of equations are then analysed for stability, requiring that the non-wetting fluid density be less that that of the wetting fluid. Example calculations in plane strain are given for a loaded vertical column, and a water injection case where a fracture face is loaded.

Fung (1990) described a control-volume finite element (CVFE) approach for coupled isothermal two-phase fluid flow and solid behaviour. The material follows a hyperbolic stress-strain law and dilatancy model (Rowe's stress dilatancy theorem) described by Agar (1984). The paper covers the assembly of the equations and the formulation in CVFEs in two dimensions. The model is verified by analysing the one-dimensional consolidation problem and comparing it to the analytical Biot solution. A two-dimensional fracture loading example is given with the presentation of stress and fluid flow changes in time. The approach appears to give robust and accurate solutions, but is limited to non-plastic material behaviour.

Scott *et al.* (1991) discussed tests on oil sands under various different conditions of shear stress to determine the permeability to water in native bitumen-saturated and extracted cores. All cores were prepared rigorously to avoid sample disturbance due to gas evolution. A large body of literature is reviewed covering permeability changes in isotropic and anisotropic loading conditions.

Schrefler and Simoni (1991) prepared the equations describing two-phase flow in a deforming porous medium — a linear momentum balance for the whole multiphase system (gas, water, and solid), and continuity equations for solid plus water and solid plus gas. Auxiliary equations included water saturation as a function of capillary pressure, the saturation constraint ($S_w + S_g = 1$), and the effective stress equation. Three combinations of solution variables were considered: $(\underline{u}, p_w, p_c)$, $(\underline{u}, p_w, p_g)$, and $(\underline{u}, p_w, S_w)$. Inertial and some convective terms were dropped in the formulation, and staggered and monolithic (simultaneous) solution schemes were tried. The best convergence was found using the combination of $(\underline{u}, p_w, p_g)$ while testing using one-dimensional consolidation, linear one-dimensional finite elements, and linear elastic solid behaviour.

Beattie *et al.* (1991) described the implementation of two essential modifications to a numerical thermal reservoir simulator to account for deformation and hysteresis of relative permeability. The authors note that the "spongy rock" approaches of contemporary simulation studies (using a large and constant pore compressibility one to two orders of magnitude greater than the measured values) leads to results conflicting with field data. The use of unrealistically high compressibilities caused the simulation pressures to steadily increase in time, while field data indicate that the injection pressure eventually levels off. In Cold Lake surface uplifts up to 45 cm had been noted, and in mature project areas (Leming pilot) a residual uplift of up to 15 cm had been recorded. A deformation model to account for the observed uplifts was devised, using two values of pore compressibility for increasing pressures, and two values for decreasing pressures. A common elastic value exists for both pressure paths, while the pore compressibility value for dilating sand was greater than that for a recompacting sand, where the dilatation recompaction initiation pressures were found using a model described in Agar (1984). The authors stress that the full integrated solution to the pore compressibility equation should be used to predict porosities as a function of pressure change. An exponential permeability function

$$k_{xyz} = k_{0_{xyz}} \exp\left[K_{MUL_{xyz}} \frac{(\phi - \phi_o)}{(1 - \phi_o)}\right] \tag{2.3}$$

was chosen to give smooth departures when the dilation pressure was traversed. A high value (100 to 500) of K_{MUL} was chosen for those blocks labelled as fracture blocks, which were activated for pressures above a predetermined failure pressure. Again, the exponential nature of the permeability relation ensured a smooth change in permeability which would not upset the convergence properties of the scheme. Relative permeability hysteresis was found to be important to match the water-oil ratios, and was applied to the water and oil only. This led to more dilation deeper into the formation. The hysteresis model ensured the oil-water relative permeabilities stayed closer to the drainage curve by using a slow scan function to the imbibition curve. The model is a useful implementation but the pore compressibility term is only pore pressure dependent, not load dependent. This will lead to errors at distances away from the wellbore, which are more pronounced after some heating has taken place.

Denbina *et al.* (1991) applied the model described by Beattie *et al.* to analyse the key reservoir drive mechanisms in the early cycles of steam stimulation at Cold Lake. Data from the mature Leming pilot project showed that there was overinjection of water (more injected than produced) in the first two cyclic steam stimulation cycles, while there was underinjection during the subsequent cycles. The authors considered the relative effects of the four mechanisms:

(i) formation compaction;

(ii) solution gas;

(iii) liquid expansion;

(iv) gravity drainage;

A fractured layer was identified, and four cycles timed by limiting pressures were simulated. As a single contributor, formation compaction was the most significant factor (> 60%) in all cycles. Solution gas was the second most important, but when combined with formation compaction it accounted for more than 80% of the recovery mechanism. Liquid expansion had a relatively minor rôle. Gravity drainage grew in importance with later cycles, presumably because of the growth of the steam chamber.

2.4 *Comments*

None of the models reviewed use plasticity to describe the material behaviour. The most common oil recovery mechanism — cyclic steam stimulation — imposes cyclic loads on the reservoir. This type of behaviour cannot adequately be modelled using non-linear elasticity, as the accumulation of irreversible strains is not accounted for. Thermal reservoir modelling with geomechanics, such as that by Beattie *et al.*, does not include the effect of loads distributed through the porous medium. A dilatory response comes from the net effective stress state, not merely the change in pore fluid pressure. The thermal loads are a key aspect in any simulation of the geomechanical response. These deficiencies in the current state of modelling are incentives for further research.

Chapter III

STATEMENT OF THE PROBLEM

Since J. Geertsma's landmark paper on the subject in 1957, the petroleum industry has realised the importance of and included rock mechanics as a component of reservoir mechanics and fracture stimulation design. In the last decade the industry has recognised that, in some cases, the behaviour of the rock or soil containing the hydrocarbon can dominate and control the recovery process. This is nowhere more true than in the thermal recovery of bitumen and heavy oil from the unconsolidated oil sands and heavy oil sands of Alberta. The objective of this study is to present the first model which combines a classical reservoir flow model with a state-of-the-art soil stress and strain model. In this manner, reservoir fluid mechanics, heat transfer mechanics, and geomechanics are permitted to interact in three dimensions as is the real situation in the thermal recovery schemes in the Athabasca oil sands and elsewhere.

3.1 *Statement of the problem*

The object of this research is to develop and demonstrate an effective three-dimensional thermal numerical reservoir simulator which incorporates the concomitant changes in stress and strain of an elastoplastic soil to hot fluid injection, such that the problem is coupled via changes in porosity and (absolute) permeability.

This objective requires the design of

a) appropriate coupled equations in three dimensions of multiphase flow, heat transfer, and solid displacements, in a form suitable for numerical solution;

b) a practical scheme to solve the coupled equations; and

c) a computer program to implement the solution.

Once the program has been developed, the model would be able to represent the events occurring as a result of hot fluid injection into unconsolidated (uncemented) sand bodies containing oil, heavy oil, or bitumen. The model will have to be able to calculate the accumulation of irrecoverable (plastic) strains, the distribution of stresses, volumetric strain, displacements, fluid pressure, temperature, and fluid saturations. This information will then lend itself to the prediction of oil recoveries from hot or cold fluid fluid injection into reservoirs where the geomechanical response is an important factor in the mechanics of recovery.

Chapter IV

DERIVATION OF THE MATHEMATICAL MODEL

The mathematical model of the physical processes occurring in three-phase flow with heat transport and solid deformations is given in this chapter. The object was to incorporate all the essential aspects of the physics while making appropriate simplifying assumptions. The aim has also been to present the mathematical model as a formulation suitable for numerical solution by both the finite element and finite difference methods.

The chapter is divided into three sections. The first describes the development of the flow and heat equations on their own. In the second section the geomechanical model is developed and coupled to the primary variables chosen for the flow and heat equations. Finally, physical property correlations are presented as continuous polynomials for use in forming the coefficients arising in the constitutive equations.

4.1 *Flow equation development: constitutive equations for heat and mass transfer*

The equation set for thermal fluid flow includes four primary variables (S_w, S_o, p_w and T). This set of constitutive equations provides unique values of the pertinent unknowns, and permits the presence of a solution gas in the bitumen, or a co-injected gas with the steam. These equations are developed without coupling to the geomechanical model so that the finite element model of the flow behaviour can be investigated independently. The coupling is detailed in the next chapter, and in fact leads to minor modifications of the following equations.

4.1.1 *Flow equations*

The general molar balance for a mixed oleic, gaseous and aqueous phase system (subscripts o, v, and w, respectively) can be expressed by (Burgess, 1978, modified):

$$\vec{\nabla}\cdot[\phi(x_{io}\rho_{om}\mathrm{v}_o S_o + x_{iw}\rho_{wm}\mathrm{v}_w S_w + y_{iv}\rho_{vm}\mathrm{v}_v S_v)] + \frac{\partial}{\partial t}[\phi(x_{io}\rho_{om}S_o + x_{iw}\rho_{wm}S_w + y_{iv}\rho_{vm}S_v)] = 0. \tag{4.1}$$

where the subscript $i = w, b, g$ for the water, bitumen and gas components, respectively. See the Nomenclature for further explanation of the symbols. Molar balances for the various components are given below by assuming that

a) bitumen does not exist in the aqueous and gaseous phases;

b) water does not exist in the oleic phase;

c) gas does not exist in the aqueous phase.

In other words, certain mole fractions are zero:

$$x_{bw} = y_{bv} = x_{wo} = x_{gw} = 0. \tag{4.2}$$

Thus we have the molar balance for bitumen:

$$\vec{\nabla}\cdot(\phi x_{bo}\rho_{om}\mathbf{v}_o S_o) + \frac{\partial}{\partial t}(\phi x_{bo}\rho_{om}S_o) = 0. \tag{4.3}$$

Molar balance for water:

$$\vec{\nabla}\cdot[\phi(x_{ww}\rho_{wm}\mathbf{v}_w S_w + y_{wv}\rho_{vm}\mathbf{v}_v S_v)] + \frac{\partial}{\partial t}[\phi(x_{ww}\rho_{wm}S_w + y_{wv}\rho_{vm}S_v)] = 0. \tag{4.4}$$

Molar balance for gas:

$$\vec{\nabla}\cdot[\phi(x_{go}\rho_{om}\mathbf{v}_o S_o + y_{gv}\rho_{vm}\mathbf{v}_v S_v)] + \frac{\partial}{\partial t}[\phi(x_{go}\rho_{om}S_o + y_{gv}\rho_{vm}S_v)] = 0. \tag{4.5}$$

where:

$$\left.\begin{aligned} x_{bo} &= \frac{\text{moles of bitumen}}{\text{total moles in oleic phase}}\\ &= \frac{\rho_{bm}}{\rho_{bm} + R_{so_r}\rho_{gm}}\\ &= \frac{\rho_{bm}}{\rho_{om}(1 + R_{so_r})}.\\ \rho_{om} &= \frac{\text{total moles of oil per unit volume of bitumen}}{\text{associated volume}}\\ &= \frac{\rho_{bm} + R_{so_r}\rho_{gm}}{1 + R_{so_r}}.\\ R_{so_r} &= R_{so}\frac{B_g}{B_b}\\ &= R_{so}\frac{\rho_{g_{sc}}\rho_b}{\rho_{b_{sc}}\rho_g}.\\ R_{so} &= \frac{\text{standard volumes of gas}}{\text{one standard volume of bitumen}}. \end{aligned}\right\} \tag{4.6}$$

Furthermo[illegible]

$$\left.\begin{aligned} x_{ww} + x_{gw} &= 1.0,\\ x_{go} + x_{bo} &= 1.0,\\ y_{wv} + y_{gv} &= 1.0, \end{aligned}\right\} \tag{4.7}$$

and by Dalton's law, $y_{wv} = p_{sat}/p_w$. The molar density, ρ'_c of component c, is related to the mass density by $\rho'_c = \rho_c/M_c$.

On converting the molar densities to mass densities, and substituting for the mole fractions, we can rewrite the equations in more familiar terms. After multiplying through by M_b the equation for bitumen becomes

$$\vec{\nabla}\cdot(\phi\hat{\rho}_b \mathbf{v}_o S_o) + \frac{\partial}{\partial t}(\phi\hat{\rho}_b S_o) = 0. \tag{4.8}$$

where

$$\hat{\rho}_b = \frac{\rho_b}{1 + R_{so_r}}. \tag{4.9}$$

After multiplying through by M_w, the equation for water becomes

$$\vec{\nabla}\cdot[\phi(\rho_w \mathbf{v}_w S_w + \alpha_v y_{wv}\rho_v \mathbf{v}_v S_v)] + \frac{\partial}{\partial t}[\phi(\rho_w S_w + \alpha_v y_{wv}\rho_v S_v)] = 0. \tag{4.10}$$

where

$$\left.\begin{aligned} \alpha_v &= \frac{M_w\rho_v}{M_v}, \\ \rho_v &= \frac{M_v p_w}{zRT}. \end{aligned}\right\} \tag{4.11}$$

After multiplying through by M_g, the equation for gas becomes

$$\vec{\nabla}\cdot[\phi(R_{so_r}\hat{\rho}_g \mathbf{v}_o S_o + \beta_v y_{gv}\rho_v \mathbf{v}_v S_v)] + \frac{\partial}{\partial t}[\phi(R_{so_r}\hat{\rho}_g S_o + \beta_v y_{gv}\rho_v S_v)] = 0. \tag{4.12}$$

where

$$\left.\begin{aligned} \hat{\rho}_g &= \frac{\rho_g}{1 + R_{so_r}}, \\ \beta_v &= \frac{M_g\rho_v}{M_v}. \end{aligned}\right\} \tag{4.13}$$

The equations for each component can now be expanded in terms of S_w, S_o, p_w and T with the use of the following information. The intrinsic phase velocity is found from Darcy's relationship:

$$\mathbf{v}_i = \frac{-\lambda_i}{\phi S_i}\mathbf{k}\cdot(\nabla p_i + \rho_i \mathbf{g}) \tag{4.14}$$

where the subscript $i = w, o, v$ for the water, oil and vapour phases, respectively, and the following constraints can be imposed:

$$\left.\begin{aligned} P_{cgw} &= p_g - p_w \\ &= 0; \\ P_{cow} &= p_o - p_w; \\ S_v + S_o + S_w &= 1. \end{aligned}\right\} \tag{4.15}$$

The zero value for the gas-water capillary pressure is only assumed here.

Products of non-linear terms AB in the equations are expanded after the first order approximation of Nolen and Berry (1972) to form expressions containing only one non-linear term:

$$A^{\nu+1}B^{\nu+1} = A^{\nu+1}B^{\nu} + A^{\nu}B^{\nu+1} - A^{\nu}B^{\nu}. \tag{4.16}$$

where the superscript ν represents a value at the old iteration level, a constant, and $\nu + 1$ is the current iteration level at which the terms act as non-linear coefficients.

The use of these constraints and expansions yields three equations in the following suitable form for discretization by the finite element method.

The water equation:

$$A_1\frac{\partial S_w}{\partial t} + A_2\frac{\partial S_o}{\partial t} + A_3\frac{\partial p_w}{\partial t} + A_4\frac{\partial T}{\partial t} + A_5 S_w + A_6 S_o + \vec{\nabla}\cdot(\mathbf{A}_7\cdot\vec{\nabla}p_w) + \vec{\nabla}\cdot\mathbf{A}_8 + A_9 = 0. \tag{4.17}$$

The bitumen equation:

$$B_1\frac{\partial S_o}{\partial t} + B_2\frac{\partial p_w}{\partial t} + B_3\frac{\partial T}{\partial t} + B_4 S_o + \vec{\nabla}\cdot(\mathbf{B}_5\cdot\vec{\nabla}p_w) + \vec{\nabla}\cdot(\mathbf{B}_6\cdot\vec{\nabla}S_o) + \vec{\nabla}\cdot\mathbf{B}_7 + B_8 = 0. \tag{4.18}$$

The gas equation:

$$C_1\frac{\partial S_w}{\partial t} + C_2\frac{\partial S_o}{\partial t} + C_3\frac{\partial p_w}{\partial t} + C_4\frac{\partial T}{\partial t} + C_5 S_w + C_6 S_o + \vec{\nabla}\cdot(\mathbf{C}_7\cdot\vec{\nabla}p_w) + \vec{\nabla}\cdot(\mathbf{C}_8\cdot\vec{\nabla}S_o) + \vec{\nabla}\cdot\mathbf{C}_9 + C_{10} = 0. \tag{4.19}$$

These equations are coupled with the formation displacements in the next section, and the resulting coefficients are fully developed in Appendix A.

4.1.2 *Energy equation*

The point conservation equation for energy, assuming kinetic energy and kinetic energy changes are small compared to those of the internal energy, U, is given by (Bird *et al.*, 1961):

$$\frac{\partial}{\partial t}(\rho U) + \vec{\nabla}\cdot(\rho\mathbf{v}U) + \vec{\nabla}\cdot\mathbf{q} - \rho(\mathbf{v}\cdot\mathbf{g}) + \vec{\nabla}\cdot(p\mathbf{v}) + \vec{\nabla}\cdot(\tau\cdot\mathbf{v}) = 0. \tag{4.20}$$

Assuming that the viscous dissipation term $\vec{\nabla}\cdot(\tau\cdot\mathbf{v})$ is negligible, and recasting the problem in terms of enthalpy, this equation becomes

$$\frac{\partial}{\partial t}(\rho h) + \vec{\nabla}\cdot(\mathbf{v}\rho h) - \frac{\partial p}{\partial t} + \vec{\nabla}\cdot\mathbf{q} - \rho(\mathbf{v}\cdot\mathbf{g}) = 0, \tag{4.21}$$

after putting $h = U + p/\rho$.

Volume averaging of the point conservation equation for energy is required because we are dealing with a multiphasic medium comprising rock, water, vapour and bitumen. This follows from the definitions given by Gray and O'Neill (1976) for an additive quantity ω distributed between two phases α and β. These are as follows. The phase average of ω in α is

$$\langle \omega_\alpha \rangle = \frac{1}{V} \int_V \omega_\alpha dV. \tag{4.22}$$

The intrinsic phase average in α is

$$\langle \omega \rangle^\alpha = \frac{1}{V_\alpha} \int_V \omega_\alpha dV. \tag{4.23}$$

Derivatives of averaged quantities are found using the transport theorem of Whitaker (1973) is

$$\left\langle \frac{\partial \omega_\alpha}{\partial t} \right\rangle = \frac{\partial \langle \omega_\alpha \rangle}{\partial t} - \frac{1}{V} \int_{A_{\alpha\beta}} \omega_\alpha \mathbf{w}_\alpha \cdot \mathbf{n}_\alpha dS. \tag{4.24}$$

Furthermore, the modified averaging theorem of Gray (1975) is

$$\langle \nabla \omega_\alpha \rangle = \epsilon_\alpha \nabla \langle \omega_\alpha \rangle^\alpha + \int_{A_{\alpha\beta}} \mathbf{n}_\alpha \hat{\omega}_\alpha dS, \tag{4.25}$$

where $A_{\alpha\beta}$ is the area of the α-β interface;
$\mathbf{w}_\alpha$ is the velocity of the α-β interface in the volume V;
$\mathbf{n}_\alpha$ is the unit normal vector along the α-β interface, outward from the α-phase;
$\hat{\omega}_\alpha$ is the deviation of ω_α from its intrinsic phase average (Gray, 1975), and
$\hat{\omega}_\alpha = \omega_\alpha - \langle \omega_\alpha \rangle^\alpha = 0$ at a point within any but the α-phase.

The point conservation equation for energy can be volume averaged. On applying the jump energy balance (Slattery, 1972)

$$\sum_{ij} \frac{1}{V} \int_{S_{ij}} [\rho_i h_i (\mathbf{v}_i - \mathbf{w}) - \rho_j h_j (\mathbf{v}_j - \mathbf{w}) - (p_i - p_j)\mathbf{w} - (\mathbf{q}_i - \mathbf{q}_j)] \cdot \mathbf{n}_{ij} dS = 0, \tag{4.26}$$

where i, j vary as the three phases, to the volume averaged equation one obtains:

$$\begin{aligned} \frac{\partial}{\partial t}\Big[(1-\phi)\rho_m h_m + \phi \sum_i S_i \rho_i h_i\Big] + \vec{\nabla} \cdot \Big[(1-\phi)\rho_m h_m \mathbf{v}_m + \phi \sum_i S_i \rho_i h_i \mathbf{v}_i\Big] - \frac{\partial}{\partial t}\Big[(1-\phi)p_m + \phi \sum_i S_i p_i\Big] \\ + \vec{\nabla} \cdot \Big[(1-\phi)\mathbf{q}_m + \phi \sum_i S_i \mathbf{q}_i\Big] - \Big[(1-\phi)\rho_m \mathbf{v}_m + \phi \sum_i S_i \rho_i \mathbf{v}_i\Big] \cdot \mathbf{g} = 0. \end{aligned} \tag{4.27}$$

If we use Fourier's heat conduction law,

$$\mathbf{q}_i = -\kappa_i \nabla T_i, \tag{4.28}$$

and assume complete and instantaneous thermal equilibrium, $T_i = T$, then we can put

$$\vec{\nabla}\cdot\left[(1-\phi)\mathbf{q}_m + \phi\sum_i S_i\mathbf{q}_i\right] = -\vec{\nabla}\cdot(\kappa\nabla T), \tag{4.29}$$

where $\kappa = (1-\phi)\kappa_m + \phi\sum_i S_i\kappa_i$.

Now, p_m is the hydrostatic stress in the medium, thus, using the tension-positive convention and using the effective stress relation with pore pressure as a positive quantity,

$$p_m = \frac{\tau_{kk}}{3} + p_w \tag{4.30}$$

If the total stress τ (see §4.2.1) is assumed to remain constant, then any change in effective stress occurs only as a result of a change in the water phase pressure, and

$$\frac{\partial p_m}{\partial t} = \frac{\partial p_w}{\partial t} \tag{4.31}$$

As a result, p_m can be set equal to p_w in Equation (4.27).

On applying the constraints given for the flow equations and the expansion of non-linear products given by Nolen and Berry (1972), an energy equation suitable for discretization is found in the following form:

$$D_1\frac{\partial S_o}{\partial t} + D_2\frac{\partial p_w}{\partial t} + D_3\frac{\partial T}{\partial t} + D_4 S_w + D_5 S_o + \mathbf{D}_6\cdot\vec{\nabla}S_o + \mathbf{D}_7\cdot\vec{\nabla}p_w + \mathbf{D}_8\cdot\vec{\nabla}T + \vec{\nabla}\cdot(D_9\nabla T) + D_{10} = 0. \tag{4.32}$$

This equation is coupled with the formation displacements in the next section, and, as for the fluid flows, the resulting coefficients are fully developed in Appendix A.

4.2 *Derivation of the solid equations and coupling*

The geomechanical model is derived using the same starting point as the classic equations of poroelasticity (Biot, 1941 and 1955). The effect of thermal expansion is introduced early in the analysis. The geomechanical model is of relatively simple form, compared to the fluid model, and is discretized in the following subsection. The discretization process allows one to separate the material behaviour into one matrix, so that an alternative material model can be used as required. This approach is merely one of convenience in the formulation, and is not a necessary condition. In this case, an elastoplastic constitutive equation is needed, and the incorporation of this behaviour is discussed.

The solid equations include strong coupling to the pore pressure and temperature. Further coupling of the primary variables through their respective equations is made possible by the analysis of the derivative of porosity with respect to time in the flow equations.

4.2.1 *Geomechanical model*

The following follows the method of Huyakorn and Pinder (1983) with the addition of thermal expansion. The geomechanical model is governed by the equilibrium equation

$$\frac{\partial \tau_{ij}}{\partial x_j} + F_i = 0. \qquad (4.33)$$

where τ_{ij} is the total stress tensor, negative in compression, and F_i is the body force per unit volume. (Note the summation convention over repeated subscripts for values of $i, j = 1, 2, 3$.) The total stress τ is separated into the Terzaghi effective stress σ' and the pore pressure p. As the grains are water-wet (i.e., the grain surface is completely surrounded by water, and the grains are not exposed to the pressures in other phases which might be present), the pore pressure is equivalent to the water phase pressure p_w:

$$\tau_{ij} = \sigma'_{ij} - \alpha' p_w \delta_{ij}. \qquad (4.34)$$

In this expression, pore pressure must be a negative quantity to be consistant with the tension-positive convention being used for stress. The Biot constant α' accounts for the difference in compressibilities between the soil skeleton and the individual grains, in the relation $\alpha' = 1 - (c_s/c_b)$ where c_s is the grain compressibility and c_b is the bulk compressibility of the skeleton. If the grains are considered incompressible, as will be the case later in this analysis, α' can be taken as one. δ_{ij} is the Kronecker delta.

After an elapsed time t, an increment of effective stress is added to the initial effective stress σ^0_{ij}:

$$\sigma'_{ij} = \sigma^0_{ij} + \Delta\sigma'_{ij}. \qquad (4.35)$$

An elastoplastic material model is considered at a later stage in the equation development. The constitutive stress-strain relationship for a linear elastic isotropic medium is:

$$\Delta\sigma'_{ij} = 2G\Delta\epsilon_{ij} + \lambda\Delta\epsilon_{kk}\delta_{ij}. \qquad (4.36)$$

where G and λ are Lamé's constants related to Young's modulus and Poisson's ratio by:

$$\left.\begin{aligned} \text{(shear modulus)} \qquad G &= \frac{E}{2(1+\nu)}, \\ \lambda &= \frac{E\nu}{(1+\nu)(1-2\nu)}. \end{aligned}\right\} \tag{4.37}$$

The strain increment $\Delta\epsilon$ includes the initial strain — the strain caused by a change in temperature ΔT,

$$\Delta\epsilon_{ij} = \epsilon_{ij} - \epsilon^0_{ij}. \tag{4.38}$$

where $\epsilon^0_{ij} = \alpha\Delta T\delta_{ij}$ (there are no shear strains associated with thermal dilation). The initial strain is actually subtracted from the total strain, of which $\Delta\epsilon$ is only the elastic component. It is convenient, however, to include the initial strain at this point. Equations (4.33) to (4.38) are combined to obtain:

$$\frac{\partial\sigma'^0_{ij}}{\partial x_j} + F_i + 2G\frac{\partial(\epsilon_{ij} - \epsilon^0_{ij})}{\partial x_j} + \lambda\frac{\partial(\epsilon_{jj} - \epsilon^0_{jj})}{\partial x_i} - \frac{\partial p_w}{\partial x_i} = 0. \tag{4.39}$$

For small deformation theory, the strain tensor is related to the displacement vector u_i by

$$\epsilon_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right). \tag{4.40}$$

Thus we obtain a governing equation for the solid skeleton:

$$\frac{\partial\sigma'^0_{ij}}{\partial x_j} + F_i + G\left(\frac{\partial^2 u_i}{\partial x_j\partial x_j} - 2\frac{\partial\epsilon^0_{ij}}{\partial x_j}\right) + (\lambda + G)\frac{\partial^2 u_j}{\partial x_i\partial x_j} - \lambda\frac{\partial\epsilon^0_{jj}}{\partial x_i} - \frac{\partial p_w}{\partial x_i} = 0. \tag{4.41}$$

where

$$\left.\begin{aligned} \frac{\partial\epsilon^0_{ij}}{\partial x_j} &= \frac{\partial(\alpha\Delta T\delta_{ij})}{\partial x_j} = \frac{\partial(\alpha\Delta T)}{\partial x_i} \\ \frac{\partial\epsilon^0_{jj}}{\partial x_i} &= \frac{\partial(3\alpha\Delta T)}{\partial x_i} \end{aligned}\right\} \tag{4.42}$$

with the boundary conditions of prescribed displacements $u_i = U_i$ on boundary portion Γ_u and prescribed surface tractions $\tau_{ij}n_j = S_{ij}$ on boundary portion Γ_s.

Approximate p_w, u_i and T using a trial function $N_J(x_j)$:

$$\left.\begin{aligned} \hat{p}(x_j,t) &= N_J(x_j)p_J(t) \\ \hat{u}_i(x_j,t) &= N_J(x_j)u_{iJ}(t) \\ \hat{T}(x_j,t) &= N_J(x_j)T_J(t) \end{aligned}\right\} \quad J = 1, 2, \ldots, n \qquad \left\{\begin{aligned} &(4.43\text{a}) \\ &(4.43\text{b}) \\ &(4.43\text{c}) \end{aligned}\right.$$

Apply the weighted residual method to Equation (4.41) with the Galerkin choice of weighting function, $W_I = N_I$, $I = 1, 2, \ldots, n$, assuming G, λ and α are constant and $\Delta\hat{T} = \hat{T} - T_0$, and integrate by parts over the domain Ω to obtain:

$$\begin{aligned}&\int_{\Omega}\left[G\frac{\partial N_I}{\partial x_j}\Big(\frac{\partial \hat{u}_i}{\partial x_j}+\frac{\partial \hat{u}_j}{\partial x_i}\Big)+\lambda\frac{\partial N_I}{\partial x_i}\frac{\partial \hat{u}_j}{\partial x_j}-\frac{\partial N_I}{\partial x_i}\hat{p}-\frac{\partial N_I}{\partial x_i}\alpha(2G+3\lambda)\hat{T}\right]d\Omega\\ &\quad-\int_{\Gamma}N_I\left[G\Big(\frac{\partial \hat{u}_i}{\partial x_j}+\frac{\partial \hat{u}_j}{\partial x_i}\Big)+\lambda\frac{\partial \hat{u}_k}{\partial x_k}\delta_{ij}-\hat{p}\delta_{ij}-\alpha(2G+3\lambda)\hat{T}\delta_{ij}+\sigma'^{0}_{ij}\right]n_j\,d\Gamma\\ &\quad=\int_{\Omega}N_I F_i\,d\Omega-\int_{\Omega}\frac{\partial N_I}{\partial x_j}\sigma'^{0}_{ij}\,d\Omega. \end{aligned}\tag{4.44}$$

Use Equations (4.35), (4.36), (4.37), (4.41), and (4.42) to obtain:

$$\tau_{ij}=\sigma'^{0}_{ij}+G\left(\frac{\partial \hat{u}_i}{\partial x_j}+\frac{\partial \hat{u}_j}{\partial x_i}\right)+\lambda\frac{\partial \hat{u}_k}{\partial x_k}\delta_{ij}-\hat{p}\delta_{ij}-(2G+3\lambda)\alpha\hat{T}\delta_{ij}. \tag{4.44}$$

Insert Equation (4.44), the approximations (4.42a), (4.42b), and (4.42c) into Equation (4.43) and use the Cauchy condition $S_i=\tau_{ij}n_j$ on Γ_s to obtain:

$$\begin{aligned}&\int_{\Omega}\left\{G\frac{\partial N_I}{\partial x_j}\Big(\frac{\partial N_J}{\partial x_j}u_{iJ}+\frac{\partial N_J}{\partial x_i}u_{jJ}\Big)+\lambda\frac{\partial N_I}{\partial x_i}\frac{\partial N_J}{\partial x_j}u_{jJ}-\frac{\partial N_I}{\partial x_i}N_J\Big[p_J+\alpha(2G+3\lambda)T_J\Big]\right\}d\Omega\\ &\quad-\int_{\Gamma_s}N_I S_i\,d\Gamma\\ &\quad=\int_{\Omega}N_I F_i\,d\Omega-\int_{\Omega}\frac{\partial N_I}{\partial x_j}\sigma'^{0}_{ij}\,d\Omega. \end{aligned}\tag{4.45}$$

The region Ω is subdivided into m finite elements Ω^e and Equation (4.45) becomes:

$$\begin{aligned}\sum_{e=1}^{m}\Bigg\{&\int_{\Omega^e}\left\{G\frac{\partial N_I}{\partial x_j}\Big(\frac{\partial N_J}{\partial x_j}u_{iJ}+\frac{\partial N_J}{\partial x_i}u_{jJ}\Big)+\lambda\frac{\partial N_I}{\partial x_i}\frac{\partial N_J}{\partial x_j}u_{jJ}-\frac{\partial N_I}{\partial x_i}N_J\Big[p_J+\alpha(2G+3\lambda)T_J\Big]\right\}d\Omega\\ &-\int_{\Gamma_s^e}N_I S_i\,d\Gamma\Bigg\}=\sum_{e=1}^{m}\left\{\int_{\Omega^e}N_I F_i\,d\Omega-\int_{\Omega^e}\frac{\partial N_I}{\partial x_j}\sigma'^{0}_{ij}\,d\Omega\right\}. \end{aligned}\tag{4.46}$$

If two additional equations are added to solve for T and p_w the following form of equilibrium equation is realized:

$$\sum_{e=1}^{m}[H_{IJ}]^e\{\chi_J\}^e=\sum_{e=1}^{m}\{R_I\}^e. \tag{4.47}$$

where

$$\{\chi_J\}^e=\begin{Bmatrix}u_{1J}\\ u_{2J}\\ u_{3J}\\ p_J\\ T_J\end{Bmatrix}\quad\text{and}\quad\{R_I\}^e=\begin{Bmatrix}R_{1I}\\ R_{2I}\\ R_{3I}\\ R_{4I}\\ R_{5I}\end{Bmatrix}.$$

It can be noted that the element stiffness matrix $[H_{IJ}]^e$ contains the element stiffness matrix and other partitioned elements:

$$\Big[H_{IJ}\Big]^e = \left[\begin{array}{c:c} [K_{IJ}] & [A_{IJ}] \\ \hdashline [A_{IJ}]^T & [-f_{IJ}] \end{array}\right]^e. \tag{4.48}$$

$[A_{IJ}]$ contains the coefficients of the pore pressure and temperature terms. $[K_{IJ}]$ corresponds to the conventional stiffness matrix in structural analysis (Zienkiewicz 1977). Thus:

$$[K_{IJ}]^e = \int_{\Omega^e} [B_I]^T [D] [B_J]\, d\Omega. \tag{4.49}$$

where the matrix $[D]$ contains only material dependent parameters—the constitutive elasticity matrix relating stress and strain—and the matrix $[B]$ relates strain to nodal displacements:

$$\{\epsilon\} = [B_J]\{u_J\}. \tag{4.50a}$$

$$\{\sigma'\} = [D]\{\epsilon - \epsilon^0\} + \{\sigma'^0\}. \tag{4.50b}$$

The constitutive matrix $[D]$ can be modified to allow for different material properties, including elastoplastic.

This analysis can be extended to an elastoplastic material by forming the elastoplasticity matrix $[D_{ep}]$. A yield condition defines the elastic limit:

$$\bar{F}(\{\sigma'\}) = \bar{\sigma}'_0. \tag{4.51}$$

where $\bar{F}$ is a scalar function of $\{\sigma'\}$ and σ'_0 is an initial yield stress.

A flow rule to determine the plastic strain increments is derived from a plastic potential $\bar{G}$:

$$\{d\epsilon^p\} = d\lambda \left\{\frac{\partial \bar{G}}{\partial \sigma}\right\} = d\bar{\epsilon}^p \left\{\frac{\partial \bar{G}}{\partial \sigma}\right\}. \tag{4.52}$$

where $d\bar{\epsilon}^p$ is the effective plastic strain increment. A material hardening or softening rule may be required:

$$\bar{F}(\{\sigma'\}) = \bar{\sigma}'(k^*). \tag{4.53}$$

where k^* is a hardening parameter. For a stationary yield surface $\bar{\sigma}'(k^*)$ is a constant. It can then be shown (Owen and Hinton, 1982) that

$$\{d\sigma'\} = [D_{ep}]\{d\epsilon - d\epsilon^0\}. \tag{4.54}$$

where

$$[D_{ep}] = [D]\left[[I] - \left\{\frac{\partial \bar{G}}{\partial \sigma}\right\}\left(\frac{\{\frac{\partial \bar{F}}{\partial \sigma}\}^{T}[D]}{S + \{\frac{\partial \bar{F}}{\partial \sigma}\}^{T}[D]\{\frac{\partial \bar{G}}{\partial \sigma}\}}\right)\right] \tag{4.55}$$

4.2.2 *Flow equations coupled to matrix motion*

To couple the solid and and fluid terms, one makes use of the relationship between the actual bulk velocity and the relative bulk velocity of each fluid, which differ in amount by the solid velocity components. The relative bulk velocity is that bulk velocity relative to the rock matrix, viz., the Darcy velocity, $\mathbf{V}_\alpha$. The actual bulk velocity is the Darcy velocity plus the rock matrix velocity, which is observed from a static frame of reference in which the motion of the rock is apparent. The actual bulk velocity is a macroscopic velocity, taken as the actual velocity, $\mathbf{v}_\alpha$, divided by the total area available for flow. Thus one can write

$$\mathbf{v}_\alpha = \frac{\mathbf{V}_\alpha}{\phi S_\alpha} + \mathbf{v}_r \tag{4.56}$$

The relative bulk velocity $\mathbf{V}_\alpha$ is the Darcy velocity in the differential form of the flow equation

$$\mathbf{V}_\alpha = -\lambda_\alpha \mathbf{k} \cdot (\nabla p_\alpha + \rho_\alpha \mathbf{g}). \tag{4.57}$$

This form of $\mathbf{v}_\alpha$ is put into the flow equations presented in §4.1.1. As the rock matrix velocity $\mathbf{v}_r$ expresses the rate of change of the displacement components $(u, w, v) \equiv (u_1, u_2, u_3)$ with time, it constitutes three of the primary variables. After Huyakorn and Pinder (1983), the solid phase continuity equation

$$\vec{\nabla} \cdot [(1-\phi)\rho_r \mathbf{v}_r] + \frac{\partial}{\partial t}[(1-\phi)\rho_r] = 0. \tag{4.58}$$

is used to eliminate the time derivative of porosity, $\partial\phi/\partial t$, from the flow equations with the substitution

$$\frac{(1-\phi)}{\rho_r}\frac{\partial \rho_r}{\partial T}\frac{\partial T}{\partial t} + \vec{\nabla} \cdot \mathbf{v}_r = \frac{\partial \phi}{\partial t} + \phi \vec{\nabla} \cdot \mathbf{v}_r. \tag{4.59}$$

Assumptions are made that the solid grains are incompressible and that substantial time derivatives $\frac{D}{\partial t}$ pertaining to $\mathbf{v}_m$ can be approximated by partial time derivatives. The three flow equations are formulated now in $S_w, S_q, p_w, T, u_1, u_2$ and u_3 and are given in Appendix A. The solid phase velocities appear during volume averaging of the point conservation equation. The modified energy equation is also given in Appendix A.

4.3 *Parametrization of fluid and rock properties*

Fluid and rock properties are commonly found during computer program execution by table look-ups. However, polynomial interpolation is necessary to make effective use of vector and parallel computers because the table look-up and interpolation is inefficient for many points, even if a single-point evaluation is quicker this way (Dubois, 1982). Non-constant rock and fluid properties have therefore been implemented as polynomials covering the entire range of anticipated pressures and temperatures.

In all cases, the appropriate polynomial approximations were found by multiple regression which both minimized the F-test value and maximized the regression correlation coefficient. Most of the data were tabulated in the literature. However, enthalpies for Athabasca bitumen were generated using published TBP and SG data (Logie *et al.*, 1978), and "PROCESS", a program by Simulation Sciences, Inc. for analyzing petroleum processing problems. The results for low pressures were in good agreement with enthalpies derived from empirical heat capacities to 300°C by Smith-Magowan *et al.* (1982), and were thus considered accurate. Tangential derivatives of properties and of their products are found as required by numerical approximation.

The biggest group of properties required for thermal reservoir simulation are those of saturated steam. The existing polynomials available in the literature are of restricted range and generally discontinuous, so a new set of interpolation polynomials are presented here and in Tortike and Farouq Ali (1989). The objective of the new work was that the polynomials should describe the behaviour of the steam properties to a sufficient degree of accuracy with changing pressure and temperature within the entire saturation envelope. The polynomials should be continuous and they should involve simple integer powers of the independent variable. Integer powers are efficient arithmetic operations, unlike the evaluation of transcendental functions, which implicitly includes fractional exponents. Transcendental functions take an order of magnitude more effort to evaluate than basic arithmetic operations on a computer. This efficiency is useful particularly in microcomputers. Simple polynomials can be most effectively evaluated using the Horner expansion (Knuth, 1981).

The data were tabulated in Perry and Green (1984) for saturated water substance. The property values were corroborated by the tables published by Cooper and Le Fevre (1975). The correlations were developed by polynomial regression, using the correlation coefficient and the residuals to judge the suitability of each correlation. Two transcendental evaluations are required for the saturation temperature and for the vapour density, and the evaluation of a square root for the specific enthalpy of vaporization. The polynomials all have integer powers.

Correlations are presented first for saturated steam condensate, then for saturated steam. Note that the use of temperature as the correlation variable results in simpler expressions than in many previously published works. If pressure is required as the independent variable in these correlations, then the evaluation of temperature from Equation (4.69) is very accurate and suitable for determining all the other properties.

The maximum and mean absolute residuals from using the correlations on the original data, expressed as percentages of the true values, are reported for the properties. The correlation coefficients are reported for the regression variables, which are sometimes transformations of the properties. The suitability of the new polynomials under particular conditions can be assessed using the residual plots in Appendix B.

A summary of the polynomial approximations is provided below, and more details regarding the degree of approximation are provided in Appendix C. Units of variables are given only the first time that they appear below.

4.3.1 *Steam condensate*

Viscosity, Pa.s, valid for $273.15\,\mathrm{K} \leq T \leq 645\,\mathrm{K}$ maximum absolute residual = 2.87%, mean absolute residual = 1.07%, and $r^2 = 0.99994$.

$$\mu_{sc} = -0.0123274 + \frac{27.1038}{T} - \frac{23527.5}{T^2} + \frac{1.01425 \times 10^7}{T^3} - \frac{2.17342 \times 10^9}{T^4} + \frac{1.86935 \times 10^{11}}{T^5}. \quad (4.60)$$

Thermal conductivity, $\mathrm{W\,m^{-1}\,K^{-1}}$, valid for $273.15\,\mathrm{K} \leq T \leq 645\,\mathrm{K}$, maximum absolute residual = 4.58%, mean absolute residual = 0.46%, and $r^2 = 0.99857$.

$$k_{hsc} = 3.51153 - 0.0443602T + 2.41233 \times 10^{-4}T^2 - 6.05099 \times 10^{-7}T^3 + 7.22766 \times 10^{-10}T^4 - 3.37136 \times 10^{-13}T^5. \quad (4.61)$$

Density, $\mathrm{kg\,m^{-3}}$, valid for $273.15\,\mathrm{K} \leq T \leq 640\,\mathrm{K}$, maximum absolute residual = 1.87%, mean absolute residual = 0.22%, and $r^2 = 0.99977$. N.B. all the absolute residuals are less than 1% for $273.15\,\mathrm{K} \leq T < 620\,\mathrm{K}$.

$$\rho_{sc} = 3786.31 - 37.2487T + 0.196246T^2 - 5.04708 \times 10^{-4}T^3 + 6.29368 \times 10^{-7}T^4 - 3.08480 \times 10^{-10}T^5. \quad (4.62)$$

Specific enthalpy, $\mathrm{kJ\,kg^{-1}}$, valid for $273.15\,\mathrm{K} \leq T \leq 645\,\mathrm{K}$, maximum absolute residual = 2.93%, mean absolute residual = 0.52%, and $r^2 = 0.99994$. N.B. all the absolute residuals are less than 1% for 305 K $< T <$ 645 K.

$$h_{sc} = 23665.2 - 366.232T + 2.26952T^2 - 0.00730365T^3 + 1.30241 \times 10^{-5}T^4 - 1.22103 \times 10^{-8}T^5 + 4.70878 \times 10^{-12}T^6. \quad (4.63)$$

4.3.2 *Steam vapour*

Viscosity, Pa.s, valid for 273.15 K $\leq T \leq$ 645 K, maximum absolute residual = 6.41%, mean absolute residual = 1.59%, and $r^2 = 0.99542$. Note that the errors are worst at the extremities of the range.

$$\mu_{sv} = -5.46807 \times 10^{-4} + 6.89490 \times 10^{-6}T - 3.39999 \times 10^{-8}T^2 + 8.29842 \times 10^{-11}T^3 - 9.97060 \times 10^{-14}T^4 + 4.71914 \times 10^{-17}T^5. \quad (4.64)$$

Thermal conductivity, W m^{-1} K^{-1} valid for 273.15 K $\leq T \leq$ 645 K, maximum absolute residual = 7.56%, mean absolute residual = 2.63%, and $r^2 = 0.99709$. Note that the errors are worst at the extremities of the range.

$$k_{hsv} = -2.35787 + 0.0297429T - 1.46888 \times 10^{-4}T^2 + 3.57767 \times 10^{-7}T^3 - 4.29764 \times 10^{-10}T^4 + 2.04511 \times 10^{-13}T^5. \quad (4.65)$$

Density, kg m^{-3}, valid for 273.15 K $\leq T \leq$ 645 K, maximum absolute residual = 7.71%, mean absolute residual = 1.29%, and $r^2 = 0.99996$ on $\ln \rho_{sv}$. Note that one can find the steam density using the real gas formula $\rho = pM/(zRT)$, assuming that the z-factor has been evaluated already.

$$\ln \rho_{sv} = -93.7072 + 0.833941T - 0.00320809T^2 + 6.57652 \times 10^{-6}T^3 - 6.93747 \times 10^{-9}T^4 + 2.97203 \times 10^{-12}T^5. \quad (4.66)$$

Specific enthalpy, kJ kg^{-1}, valid for 273.15 K $\leq T \leq$ 640 K, maximum absolute residual = 0.50%, mean absolute residual = 0.08%, and $r^2 = 0.99920$,

$$h_{sv} = -22026.9 + 365.317T - 2.25837T^2 + 0.00737420T^3 - 1.33437 \times 10^{-5}T^4 + 1.26913 \times 10^{-8}T^5 - 4.96880 \times 10^{-12}T^6. \quad (4.67)$$

4.3.3 *Steam vapour and condensate*

Specific enthalpy of vaporization (latent heat), kJ kg^{-1}, valid for $273.15\,\mathrm{K} \leq T \leq 645\,\mathrm{K}$, maximum absolute residual = 5.68%, mean absolute residual = 0.40%, and $\mathrm{r}^2 = 0.99994$ on L_v^2. Note that all the absolute residuals are less than 1% for $273.15\,\mathrm{K} \leq T < 620\,\mathrm{K}$.

$$L_v = \left(7184500 + 11048.6T - 88.4050T^2 + 0.162561T^3 - 1.21377 \times 10^{-4}T^4\right)^{\frac{1}{2}}. \tag{4.68}$$

Saturation temperature, K, valid for $0.611\,\mathrm{kPa} \leq p \leq 22.12\,\mathrm{MPa}$, maximum absolute residual = 0.11%, mean absolute residual = 0.03%, and $\mathrm{r}^2 = 1.00000$. Note that the pressure range corresponds to $273.15\,\mathrm{K} \leq T \leq 647.3\,\mathrm{K}$.

$$T = 280.034 - 14.0856\ln p + 1.38075(\ln p)^2 - 0.101806(\ln p)^3 + 0.0190017(\ln p)^4. \tag{4.69}$$

Saturation pressure, kPa, valid for $280\,\mathrm{K} \leq T \leq 647.3\,\mathrm{K}$, maximum absolute residual = 3.22%, mean absolute residual = 0.49%, and $\mathrm{r}^2 = 1.00000$ on $\sqrt{p}$. Note that all the absolute residuals are less than 1% for $320\,\mathrm{K} \leq T \leq 647.3\,\mathrm{K}$.

$$p = \Big(-175.776 + 2.29272T - 0.0113953T^2 + 2.62780 \times 10^{-5}T^3 - 2.73726 \times 10^{-8}T^4 + 1.13816 \times 10^{-11}T^5\Big)^2. \tag{4.70}$$

4.3.4 *Bitumen*

Enthalpy, J kg^{-1}, obtained from data generated by the Simulation Science "PROCESS" program.

$$h_b = a + bT + cT^2 + dp + ep^2. \tag{4.71}$$

$$273\,\mathrm{K} \leq T \leq 625\,\mathrm{K} \quad \text{and} \quad 0.1\,\mathrm{MPa} \leq p \leq 21\,\mathrm{MPa}$$

4.3.5 *Gas (methane)*

Solution gas ratio, vol/vol, taken from Svrcek and Mehrotra (1982). A suitable multiple regression was chosen with some extrapolation.

$$R_{so} = a + bp + c\frac{p}{T}. \tag{4.72}$$

Viscosity, Pa.s (Perry and Green, 1984)

$$\mu_g = a + bT. \qquad (4.73)$$
$$250\,\mathrm{K} \leq T \leq 600\,\mathrm{K}$$

Thermal conductivity, W m^{-1} K^{-1} (Perry and Green, 1984)

$$\kappa_g = a + bT + cT^2. \qquad (4.74)$$
$$250\,\mathrm{K} \leq T \leq 600\,\mathrm{K}$$

Enthalpy, J kg^{-1} (Perry and Green, 1984). Several multiple regressions were investigated:

$$h_g = a + bT + cp + dpT. \qquad (4.75)$$
$$250\,\mathrm{K} \leq T \leq 500\,\mathrm{K} \quad \text{and} \quad 0.1\,\mathrm{MPa} \leq p \leq 30\,\mathrm{MPa}$$

for which $\mathrm{r}^2 = 0.996$.

4.3.6 *Rock matrix*

The properties of the oil sand fabric were found as a composite of six chemical components: calcite, aragonite, quartz, pyrite, albite and microcline. The individual constituent properties are found in Touloukian *et al.*(1970) and can be combined using the appropriate average.

Density, kg m^{-3}

$$\rho = \rho_o(1 + \alpha_{T_i}(T - T_o)); \qquad (4.76a)$$

and

$$\rho_o = \sum_{i=1}^{6} \rho_i v_i. \qquad (4.76b)$$

where v_i is the volume fraction of component i.

Coefficient of thermal expansion, K^{-1}

$$\alpha_T = \sum_{i=1}^{6} \alpha_{T_i} v_i. \qquad (4.77)$$

Thermal conductivity, W m^{-1} K^{-1}

$$\kappa = \prod_{i=1}^{6} \kappa_i^{v_i}. \tag{4.78}$$

Heat capacity, J kg^{-1} K^{-1}

$$c_p = T \sum_{i=1}^{6} b_i + \sum_{i=1}^{6} a_i. \tag{4.79}$$

4.3.7 *Rock-fluid properties*

The relative permeability data for Athabasca tar sand was taken from Tortike (1984):

$$k_{rg} = a + bS_\ell + cS_\ell^2 + dS_\ell^3, \tag{4.80a}$$

$$k_{rw} = a + bS_w + cS_w^2 + dS_w^3, \tag{4.80b}$$

$$k_{r_{ow}} = a + bS_w + cS_w^2 + dS_w^3 + eS_w^4 + fS_w^5, \tag{4.80c}$$

$$k_{r_{og}} = a + bS_\ell + cS_\ell^2 + dS_\ell^3, \tag{4.80d}$$

$$P_{c_{ow}} = a + bS_w + cS_w^2 + dS_w^{-1}, \tag{4.80e}$$

Three phase oil relative permeabilities are calculated from Stone's modified method II (Aziz and Settari, 1973), which is given below:

$$k_{ro} = k_{rocw} \left[(k_{rw} + \frac{k_{row}}{k_{rocw}})(k_{rg} + \frac{k_{rog}}{k_{rocw}}) - (k_{rw} + k_{rg}) \right]. \tag{4.81}$$

where

$$k_{row}(S_{wc}) = k_{rog}(S_L = 1) = k_{rocw}$$

4.3.8 *Gas law deviation factors, z*

The original Redlich-Kwong equation of state can be formulated as a cubic equation of state in z (Reid et al., 1977). A value for z can be then found using Cardano's formula for solving a cubic polynomial.

$$z^3 - z^2 + (A^* - B^{*2} - B^*)z - A^*B^* = 0. \tag{4.82}$$

where

$$A^* = \frac{\Omega_a p_r}{T_r^2} \quad \text{and} \quad B^* = \frac{\Omega_b p_r}{T_r}.$$

and

$$\Omega_a = \frac{1}{9 \times 2^{1/3} - 1} \quad \text{and} \quad \Omega_b = \frac{2^{1/3} - 1}{3}$$

It is recognized that steam is a polar compound and therefore there will be some error in the prediction of z for the methane-steam mixtures.

4.3.9 *Mixture properties*

The mixing rules for the Redlich-Kwong equation of state are (Reid *et al.*, 1977)

$$A^*_m = \left(\sum_j y_j A_j^{*\,1/2}\right)^2 ; \tag{4.83a}$$

$$B^*_m = \sum_j y_j B^*_j ; \tag{4.83b}$$

The gaseous phase (vapour) in the model is composed of methane and steam. The densities for pure methane and the vapour can be found from z

$$\rho = \frac{pM}{zRT} \tag{4.84}$$

The thermal conductivity of the binary-composition vapour can be found from the Brokaw empirical method (Reid *et al.*, 1977), where

$$\left.\begin{aligned} \kappa_m &= q\kappa_L + (1-q)\kappa_R; \\ \kappa_L &= y_1\kappa_1 + y_2\kappa_2; \\ \frac{1}{\kappa_R} &= \frac{y_1}{\kappa_1} + \frac{y_2}{\kappa_2}; \end{aligned}\right\} \tag{4.85}$$

The value of q is arbitrarily set to 0.5. The viscosity of the vapour mixture can be estimated using Chapman-Enskog kinetic theory (Reid *et al.*, 1977), so that for a binary mixture at low pressure:

$$\mu_m = \frac{y_1\mu_1}{y_1 + y_2\phi_{12}} + \frac{y_2\mu_2}{y_2 + y_1\phi_{21}}. \tag{4.86}$$

The interaction parameters ϕ_{ij} can be found using the approximation of Herning and Zipperer (Reid *et al.*, 1977)

$$\phi_{ij} = \left(\frac{M_j}{M_i}\right)^{1/2} = \frac{1}{\phi_{ji}}. \tag{4.87}$$

The oleic phase is composed of methane dissolved in various concentrations in the bitumen. The thermal conductivity of this mixture can be found using a power law relation described by

Vredeveld (Reid et al., 1977):

$$\kappa_m^r = w_1\kappa_1^r + w_2\kappa_2^r. \tag{4.88}$$

where w_i is the mass fraction of component i in the mixture, and r is set to -2 if $1 \leq \frac{\kappa_2}{\kappa_1} \leq 2$. The mass fraction is easily obtained from the mole fraction as

$$w_1 = \frac{x_1 M_1}{(1 - x_1)M_2 + x_1 M_1}. \tag{4.89}$$

Svrcek and Mehrotra (1982) have investigated the viscosity of bitumen containing dissolved methane. They present their results in the Walther-type equation:

$$\log_{10}[\log_{10}(\mu)] = -7.515w_1 + 0.78872 - 0.004471T. \tag{4.90}$$

where w_1 is the mass fraction of methane in the mixture, μ is in mPa.s, and the temperature T is in °C.

Chapter V

DISCRETIZATION AND SOLUTION OF THE EQUATIONS — COUPLED MATHEMATICAL MODEL FOR FINITE ELEMENTS

The mathematical model derived in the first two sections of Chapter IV is a set of partial differential equations which cannot be solved analytically. Furthermore, there are strongly non-linear coefficients in the equations, which imply that not only must the equations be prepared for numerical solution, but there must be applied a method of linearization to the resultant system of equations. In this work the method of weighted residuals with Petrov-Galerkin weighting functions has been applied to the original equations of Chapter IV to prepare them for solution by the finite element method. The discretizations have been performed separately in the two solution domains of space, Ω, and time, t. A continuous solution is found in the space domain, while a marching scheme provides the solution in the temporal domain. The discretizations in both domains are brought together to be linearized by the Newton-Raphson method. Certain details of the linearization are examined closely to help the final implementation of the solution as a computer program. The implementation was never able to provide a satisfactory convergent solution, which is discussed in the final section of this chapter.

5.1 *Spatial integration of a general non-linear partial differential equation*

The basic method for spatial integration of a general linear partial differential equation can be found in Voss (1978) and in Zienkiewicz and Morgan (1983). Here the development is shown for the particular differential equations developed in the previous chapter. Index notation is used where it is more compact, and it may be assumed whenever a summation symbol is not present. Appendix C should be consulted for more information on the method of weighted residuals and finite elements.

Consider the region of three-dimensional space, Ω, bounded by the closed curve Γ, and a governing equation

$$\mathfrak{D}(\psi(\Omega, t)) = 0. \tag{5.1}$$

where $\mathfrak{D}$ is some non-linear operator and ψ is some unknown function in Ω and time. The boundary conditions are included in this expression. We can introduce a typical non-linear coefficient, $C(\Omega, t, \psi)$, which is dependent on space, time and ψ. Suppose there exists a set of independent trial functions, $N_j(\Omega)$. This set of trial, or basis, functions can be used to approximate the variation of

$C(\Omega, t, \psi)$ and $\psi(\Omega, t)$ in the region Ω. Then we can use $N_j(\Omega) : j = 1, 2, 3, \ldots, n$ as a set of basis functions in terms of which $C(\Omega, t, \psi)$ and $\psi(\Omega, t)$ can be approximated in space:

$$\psi(\Omega, t) \approx \hat{\psi}(\Omega, t) = \psi_j(t)\, N_j(\Omega). \tag{5.2}$$

$$C(\Omega, t, \psi) \approx \hat{C}(\Omega, t, \psi) = c_j(t, \psi)\, N_j(\Omega). \tag{5.3}$$

where the variable with the hat is an approximation to the true value of the function.

Now the weighted residual statement is:

$$\int_\Omega R W_i(\Omega)\, d\Omega = \int_\Omega \mathfrak{D}(\hat{\psi}(\Omega, t)) W_i(\Omega)\, d\Omega = 0. \tag{5.4}$$

The question arises as to what terms are contained under the non-linear differential operator $\mathfrak{D}$. Development of the flow and heat transfer equations in the previous chapter, also given in Appendix A, showed that the following seven specimen terms appear:

$$\vec{\nabla} \cdot \vec{A}(\Omega, t, \psi)\psi(\Omega, t); \tag{5.5a}$$

$$B(\Omega, t, \psi)\frac{\partial \psi(\Omega, t)}{\partial t}; \tag{5.5b}$$

$$C(\Omega, t, \psi)\psi(\Omega, t); \tag{5.5c}$$

$$\vec{\nabla} \cdot \left[\vec{\vec{D}}(\Omega, t, \psi) \cdot \vec{\nabla}\psi(\Omega, t)\right]; \tag{5.5d}$$

$$\vec{E}(\Omega, t, \psi) \cdot \vec{\nabla}\psi(\Omega, t); \tag{5.5e}$$

$$F(\Omega, t, \psi); \tag{5.5f}$$

$$\vec{\nabla} \cdot \vec{G}(\Omega, t, \psi). \tag{5.5g}$$

Each term can be approximated using the trial functions $N_j(\psi)$ as in Equations (5.2) and (5.3). Each term will fall under the integral of the weighted residual statement given in Equation (5.4), after being multiplied by the weighting function, $W_i(\Omega)$. Each term can be integrated separately in the order they appeared in Equations (5.5a) to (5.5g), and will appear as follows:

$$\int_\Omega \vec{a}_k(t, \psi) \cdot \vec{\nabla} N_k(\Omega)\, \psi_j(t) N_j(\Omega) W_i(\Omega)\, d\Omega = \psi_j \vec{a}_k \cdot \int_\Omega \vec{\nabla} N_k N_j W_i\, d\Omega. \tag{5.6}$$

$$\int_\Omega b_k(t, \psi) N_k(\Omega)\, \frac{d\psi_j(t)}{dt} N_j(\Omega) W_i(\Omega)\, d\Omega = \frac{d\psi_j(t)}{dt} b_k \int_\Omega N_k N_j W_i\, d\Omega. \tag{5.7}$$

$$\int_\Omega c_k(t, \psi) N_k(\Omega)\, \psi_j(t) N_j(\Omega) W_i(\Omega)\, d\Omega = \psi_j(t) c_k \int_\Omega N_k N_j W_i\, d\Omega. \tag{5.8}$$

Before expanding the next term in terms of the approximation, we must use Green's lemma to extract the boundary integral. Green's lemma in three dimensions is (Zienkiewicz, 1977):

$$\int_\Omega \phi \vec{\nabla}\psi \, d\Omega = -\int_\Omega \vec{\nabla}\phi\psi \, d\Omega + \int_\Gamma \phi\psi\vec{n} \, d\Gamma. \tag{5.9}$$

Thus we can write:

$$\begin{aligned} \int_\Omega \vec{\nabla} \cdot [\vec{\vec{D}}(\Omega, t, \psi) \cdot \vec{\nabla}\psi(\Omega, t)] W_i(\Omega) \, d\Omega &= \int_\Gamma (\vec{\vec{D}} \cdot \vec{\nabla}\psi) \cdot \vec{n} W_i \, d\Gamma - \int_\Omega (\vec{\vec{D}} \cdot \vec{\nabla}\psi) \cdot \vec{\nabla} W_i \, d\Omega \\ &= \int_\Gamma (\vec{\vec{D}} \cdot \vec{\nabla}\psi) \cdot \vec{n} W_i \, d\Gamma - \int_\Omega (\vec{\vec{d}}_k N_k : \psi_j \vec{\nabla} N_j) \cdot \vec{\nabla} W_i \, d\Omega \\ &= \int_\Gamma (\vec{\vec{D}} \cdot \vec{\nabla}\psi) \cdot \vec{n} W_i \, d\Gamma - \psi_j \vec{\vec{d}}_k : \int_\Omega N_k \vec{\nabla} N_j \vec{\nabla} W_i \, d\Omega \end{aligned} \tag{5.10}$$

$$\int_\Omega \vec{e}_k(t, \psi) N_k(\Omega) \cdot \psi_j(t) \vec{\nabla} N_j(\Omega) W_i(\Omega) \, d\Omega = \psi_j \vec{e}_k \cdot \int_\Omega N_k \vec{\nabla} N_j W_i \, d\Omega. \tag{5.11}$$

$$\int_\Omega f_j(t, \psi) N_j(\Omega) W_i(\Omega) \, d\Omega = f_j \int_\Omega N_j W_i \, d\Omega. \tag{5.12}$$

$$\int_\Omega \vec{g}_j(t, \psi) \cdot \vec{\nabla} N_j(\Omega) W_i(\Omega) \, d\Omega = \vec{g}_j \cdot \int_\Omega \vec{\nabla} N_j W_i \, d\Omega. \tag{5.13}$$

All the subscripts i, j, and k vary from 1 to n, and repeated subscripts indicate summation. Now all the coefficients of ψ_j and $\frac{d\psi_j}{dt}$ can be separately collected, as can all the remaining terms, to be represented in three, general, non-linear coefficients:

$$S_{ij}(\psi, t)\psi_j(t) + D_{ij}(\psi, t)\frac{d\psi_j(t)}{dt} + F_i(\psi, t) = 0. \tag{5.14}$$

where the repeated subscripts here indicate summation. This is just a rearrangement of the weighted residual statement of Equation (4.4) using the seven typical non-linear terms arising from the governing equations of the previous chapter.

In the Galerkin approach, the weighting functions $W_i(\Omega)$ appearing in the seven typical terms would be set equal to the trial functions $N_i(\Omega)$. A variation is the Petrov-Galerkin method, where some weighting functions conform to the Galerkin approach, but others have different values from the trial functions.

5.1.1 *Upwinding and the Petrov-Galerkin method*

A convection equation can be represented by (Voss, 1978):

$$\frac{\partial\psi}{\partial t} + \vec{v} \cdot \vec{\nabla}\psi = 0. \tag{5.15}$$

The solution to this equation at any point in space is influenced by events upstream. Thus, a centered-in-space approximation is unlikely to be accurate. The Galerkin method assigns equal influence from surrounding nodes on a given node, which will inadequately describe a convective term. In a mixed equation, containing convective and non-convective terms, the exclusive use of the Galerkin weighting functions with convective terms will be inaccurate. Instead, a separate set of weighting functions can be chosen for the convective terms which attribute more influence to those nodes upstream of the point in question. Determining where is upstream and downstream of a point is done by local inspection of the velocity term. The process of ascribing more influence to upstream nodes is called upstream weighting, or simply, upwinding.

Typical convective terms are $\vec{v} \cdot \vec{\nabla}\psi$ and $\vec{\nabla} \cdot \vec{c}\psi$, where ψ is the property being convected with a velocity v. The convective term is not always immediately obvious, such as when the velocity is not explicitly found in the expression. The asymmetric weighting function chosen to facilitate upwinding is described in Appendix C. The mixed use of the trial functions as the weighting functions, and asymmetric weighting functions, together in the same weighted residual statement is referred to as the Petrov-Galerkin choice of weighting functions.

5.1.2 *Choice of finite elements*

Two different finite elements have been chosen for this work. There is an eight-noded brick element, the L8 element, and a six-noded triangular prism element, the L6 element. The elements and their isoparametric coordinates are illustrated in Figure 5–1. Both elements are linear elements, i.e., they implement linear interpolation functions for the degrees of freedom—hence the L in L6 and L8. Both types of element are isoparametric, which means that the element geometry and the approximated variables are represented by the same basis functions. Furthermore, it permits solutions to be found within a local coordinate system, convenient for the numerical integration, which are mapped to the global coordinate system by the linear functions used for interpolation. This improves the spatial discretization because we no longer rely on regular geometries to represent curved boundaries or to implement a locally-refined mesh. Refer to Appendix C for more detailed information about element mapping and numerical integration.

Two element geometries are provided by the L6 and L8 finite elements. Normally, different finite elements are used to provide trial functions of a higher degree. The purpose here is to provide flexibility when defining the geometry of possible formation parting planes. The triangular prism element allows a potential inclining parting plane to be represented in the mesh. It does this in conjunction with the brick elements which are also non-regular in the global coordinate system. The

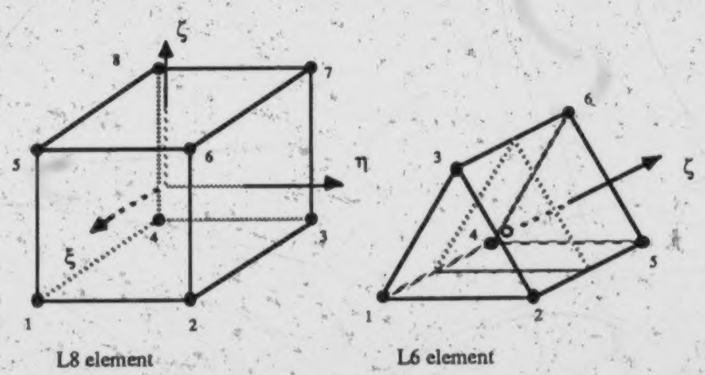


Figure 5–1

Finite elements used by FESPS.

L6 elements also can improve mesh resolution easily.

5.2 *General numerical solution of the equations*

The equations for the molar balances, the energy balance, and the displacements have been posed in a manner suitable for discretization. The equations are now discretized in space using the method of weighted residuals to yield an equation for a finite element. The element equation is then discretized in space according to the general scheme described in Appendix C. The resulting general equation is then ready to be solved for a given set of initial and boundary conditions. It is, however, a non-linear equation and must be linearized to facilitate the solution. The Newton-Raphson method is used to linearize the general equation posed as $\mathfrak{M} = 0$. The coefficient matrix of this set of linear equations is the Jacobian matrix—a matrix of derivatives of the function $\mathfrak{M}$. The derivatives are first reduced to selected components and then formed numerically. The reduction step requires algebraic manipulation and reduces the computational requirements. The following sections cover this material in detail.

5.3 *Application of the method of weighted residuals and expansion by trial functions*

The method of weighted residuals (MWR) is applied to each physical equation. The polynomial order of the basis functions can be reduced using finite subdomains for a given solution accuracy at specified points. The basis functions are only required to be continuous in certain subdomains—finite elements—so that the approximations are piecewise continuous in the entire solution domain. Thus we arrive at individual element equations for each degree of freedom. The integrated terms arising from application of the MWR and piecewise continuous basis functions must be expressed in terms of the basis functions and nodal values. The nodes define the element geometry and the polynomial order of the basis functions.

Thus the physical equations have been discretized in space. The degrees of freedom as functions of time and space are reduced to terms which vary at each node only as a function of time, while the basis functions vary with space. The following pages show how each term falling under the integration can be expanded, for each physical equation and a given element.

Note the following convention on the use of indices:

a) $H, I, J, K = [1..\mathbf{nnodes}]$ are used as nodal indices ($\mathbf{nnodes} = 8$);

b) $i, j, k = [1..3]$ are used as spatial dimension indices;

c) $\alpha, \beta, \gamma = [1..\mathbf{ndof}]$ are used to indicate particular degrees of freedom or physical equations (water, bitumen, gas, energy, displacements).

5.3.1 *Water equation*

Expansion of integrals by term:

i) $$\int_{\Omega^e} A_1 \frac{\partial S_w}{\partial t} N_I \, d\Omega = \sum_{J=1}^{n} \frac{\partial S_{w_J}}{\partial t} \sum_{K=1}^{n} A_{1_K} \int_{\Omega^e} N_K N_J N_I \, d\Omega.$$

ii) $$\int_{\Omega^e} A_2 \frac{\partial S_o}{\partial t} N_I \, d\Omega = \sum_{J=1}^{n} \frac{\partial S_{o_J}}{\partial t} \sum_{K=1}^{n} A_{2_K} \int_{\Omega^e} N_K N_J N_I \, d\Omega.$$

iii) $$\int_{\Omega^e} A_3 \frac{\partial p_w}{\partial t} N_I \, d\Omega = \sum_{J=1}^{n} \frac{\partial p_{w_J}}{\partial t} \sum_{K=1}^{n} A_{3_K} \int_{\Omega^e} N_K N_J N_I \, d\Omega.$$

iv) $$\int_{\Omega^e} A_4 \frac{\partial T}{\partial t} N_I \, d\Omega = \sum_{J=1}^{n} \frac{\partial T_J}{\partial t} \sum_{K=1}^{n} A_{4_K} \int_{\Omega^e} N_K N_J N_I \, d\Omega.$$

v) $$\int_{\Omega^e} A_5 S_w N_I \, d\Omega = \sum_{J=1}^{n} S_{w_J} \sum_{K=1}^{n} A_{5_K} \int_{\Omega^e} N_K N_J N_I \, d\Omega.$$

vi) $$\int_{\Omega^e} A_6 S_o N_I \, d\Omega = \sum_{J=1}^{n} S_{o_J} \sum_{K=1}^{n} A_{6_K} \int_{\Omega^e} N_K N_J N_I \, d\Omega.$$

vii) $$\int_{\Omega^e} \vec{\nabla} \cdot \left(\vec{\vec{A}}_7 \cdot \vec{\nabla} p_w \right) W_I \, d\Omega = \int_B \left(\vec{\vec{A}}_7 \cdot \vec{\nabla} p_w \right) W_I \cdot \vec{n} \, dB - \sum_{J=1}^{n} p_{w_J} \sum_{K=1}^{n} \vec{\vec{A}}_{7_K} : \int_{\Omega^e} N_K \vec{\nabla} N_J \vec{\nabla} W_I \, d\Omega.$$

viii) $$\int_{\Omega^e} \vec{\nabla} \cdot \vec{A}_8 W_I \, d\Omega = \sum_{J=1}^{n} \vec{A}_{8_J} \cdot \int_{\Omega^e} \vec{\nabla} N_J W_I \, d\Omega.$$

ix) $$\int_{\Omega^e} A_9 N_I \, d\Omega = \sum_{J=1}^{n} A_{9_J} \int_{\Omega^e} N_J N_I \, d\Omega.$$

x) $$\int_{\Omega^e} A_{10} \frac{\partial}{\partial t} (\vec{\nabla} \cdot \vec{u}) N_I \, d\Omega = \sum_{i=1}^{3} \sum_{J=1}^{n} \frac{\partial u_{i_J}}{\partial t} \sum_{K=1}^{n} A_{10_K} \int_{\Omega^e} N_K \frac{\partial N_J}{\partial x_i} N_I \, d\Omega.$$

Where any continuous primary variable $\psi(\Omega, t)$ is a function of time t and space Ω; the value of a primary variable at a node J is $\psi_J(t)$, a function of time only; the trial function $N_I(\Omega)$ a function of space only; $n = 8$ is the number of nodal values; $\vec{u} \equiv u_i = (u_1, u_2, u_3)$ is the displacement vector; $\frac{\partial \vec{u}}{\partial t}$ is the velocity of the solid matrix; and $\vec{n}$ is the normal to the boundary surface Γ.

5.3.2 *Bitumen (oil) equation*

Expansion of integrals by term:

i)
$$\int_{\Omega^e} B_1 \frac{\partial S_o}{\partial t} N_I \, d\Omega = \sum_{J=1}^{n} \frac{\partial S_{o_J}}{\partial t} \sum_{K=1}^{n} B_{1_K} \int_{\Omega^e} N_K N_J N_I \, d\Omega.$$

ii)
$$\int_{\Omega^e} B_2 \frac{\partial p_w}{\partial t} N_I \, d\Omega = \sum_{J=1}^{n} \frac{\partial p_{w_J}}{\partial t} \sum_{K=1}^{n} B_{2_K} \int_{\Omega^e} N_K N_J N_I \, d\Omega.$$

iii)
$$\int_{\Omega^e} B_3 \frac{\partial T}{\partial t} N_I \, d\Omega = \sum_{J=1}^{n} \frac{\partial T_J}{\partial t} \sum_{K=1}^{n} B_{3_K} \int_{\Omega^e} N_K N_J N_I \, d\Omega.$$

iv)
$$\int_{\Omega^e} B_4 S_o N_I \, d\Omega = \sum_{J=1}^{n} S_{o_J} \sum_{K=1}^{n} B_{4_K} \int_{\Omega^e} N_K N_J N_I \, d\Omega.$$

v)
$$\int_{\Omega^e} \vec{\nabla} \cdot \left(\vec{\vec{B}}_5 \cdot \vec{\nabla} p_w \right) W_I \, d\Omega = \int_B \left(\vec{\vec{B}}_5 \cdot \vec{\nabla} p_w \right) W_I \cdot \vec{n} \, dB$$
$$- \sum_{J=1}^{n} p_{w_J} \sum_{K=1}^{n} \vec{\vec{B}}_{5_K} : \int_{\Omega^e} N_K \vec{\nabla} N_J \vec{\nabla} W_I \, d\Omega.$$

vi)
$$\int_{\Omega^e} \vec{\nabla} \cdot \left(\vec{\vec{B}}_6 \cdot \vec{\nabla} S_o \right) W_I \, d\Omega = \int_B \left(\vec{\vec{B}}_6 \cdot \vec{\nabla} S_o \right) W_I \cdot \vec{n} \, dB$$
$$- \sum_{J=1}^{n} S_{o_J} \sum_{K=1}^{n} \vec{\vec{B}}_{6_K} : \int_{\Omega^e} N_K \vec{\nabla} N_J \vec{\nabla} W_I \, d\Omega.$$

vii)
$$\int_{\Omega^e} \vec{\nabla} \cdot \vec{B}_7 W_I \, d\Omega = \sum_{J=1}^{n} \vec{B}_{7_J} \cdot \int_{\Omega^e} \vec{\nabla} N_J W_I \, d\Omega.$$

viii)
$$\int_{\Omega^e} B_8 N_I \, d\Omega = \sum_{J=1}^{n} B_{8_J} \int_{\Omega^e} N_J N_I \, d\Omega.$$

ix)
$$\int_{\Omega^e} B_9 \frac{\partial}{\partial t} (\vec{\nabla} \cdot \vec{u}) N_I \, d\Omega = \sum_{i=1}^{3} \sum_{J=1}^{n} \frac{\partial u_{i_J}}{\partial t} \sum_{K=1}^{n} B_{9_K} \int_{\Omega^e} N_K \frac{\partial N_J}{\partial x_i} N_I \, d\Omega.$$

Where any continuous primary variable $\psi(\Omega, t)$ is a function of time t and space Ω; the value of a primary variable at a node J is $\psi_J(t)$, a function of time only; the trial function $N_I(\Omega)$ a function of space only; $n = 8$ is the number of nodal values; $\vec{u} \equiv u_i = (u_1, u_2, u_3)$ is the displacement vector; $\frac{\partial \vec{u}}{\partial t}$ is the velocity of the solid matrix; and $\vec{n}$ is the normal to the boundary surface Γ.

5.3.3 *Gas equation*

Expansion of integrals by term:

i) $$\int_{\Omega^e} C_1 \frac{\partial S_w}{\partial t} N_I \, d\Omega = \sum_{J=1}^{n} \frac{\partial S_{w_J}}{\partial t} \sum_{K=1}^{n} C_{1_K} \int_{\Omega^e} N_K N_J N_I \, d\Omega.$$

ii) $$\int_{\Omega^e} C_2 \frac{\partial S_o}{\partial t} N_I \, d\Omega = \sum_{J=1}^{n} \frac{\partial S_{o_J}}{\partial t} \sum_{K=1}^{n} C_{2_K} \int_{\Omega^e} N_K N_J N_I \, d\Omega.$$

iii) $$\int_{\Omega^e} C_3 \frac{\partial p_w}{\partial t} N_I \, d\Omega = \sum_{J=1}^{n} \frac{\partial p_{w_J}}{\partial t} \sum_{K=1}^{n} C_{3_K} \int_{\Omega^e} N_K N_J N_I \, d\Omega.$$

iv) $$\int_{\Omega^e} C_4 \frac{\partial T}{\partial t} N_I \, d\Omega = \sum_{J=1}^{n} \frac{\partial T_J}{\partial t} \sum_{K=1}^{n} C_{4_K} \int_{\Omega^e} N_K N_J N_I \, d\Omega.$$

v) $$\int_{\Omega^e} C_5 S_w N_I \, d\Omega = \sum_{J=1}^{n} S_{w_J} \sum_{K=1}^{n} C_{5_K} \int_{\Omega^e} N_K N_J N_I \, d\Omega.$$

vi) $$\int_{\Omega^e} C_6 S_o N_I \, d\Omega = \sum_{J=1}^{n} S_{o_J} \sum_{K=1}^{n} C_{6_K} \int_{\Omega^e} N_K N_J N_I \, d\Omega.$$

vii) $$\int_{\Omega^e} \vec{\nabla} \cdot \left(\vec{\vec{C}}_7 \cdot \vec{\nabla} p_w \right) W_I \, d\Omega = \int_B \left(\vec{\vec{C}}_7 \cdot \vec{\nabla} p_w \right) W_I \cdot \vec{n} \, dB$$
$$- \sum_{J=1}^{n} p_{w_J} \sum_{K=1}^{n} \vec{\vec{C}}_{7_K} : \int_{\Omega^e} N_K \vec{\nabla} N_J \vec{\nabla} W_I \, d\Omega.$$

viii) $$\int_{\Omega^e} \vec{\nabla} \cdot \left(\vec{\vec{C}}_8 \cdot \vec{\nabla} S_o \right) W_I \, d\Omega = \int_B \left(\vec{\vec{C}}_8 \cdot \vec{\nabla} S_o \right) W_I \cdot \vec{n} \, dB$$
$$- \sum_{J=1}^{n} S_{o_J} \sum_{K=1}^{n} \vec{\vec{C}}_{8_K} : \int_{\Omega^e} N_K \vec{\nabla} N_J \vec{\nabla} W_I \, d\Omega.$$

ix) $$\int_{\Omega^e} \vec{\nabla} \cdot \vec{C}_9 W_I \, d\Omega = \sum_{J=1}^{n} \vec{C}_{9_J} \cdot \int_{\Omega^e} \vec{\nabla} N_J W_I \, d\Omega.$$

x) $$\int_{\Omega^e} C_{10} N_I \, d\Omega = \sum_{J=1}^{n} C_{10_K} \int_{\Omega^e} N_J N_I \, d\Omega.$$

xi) $$\int_{\Omega^e} C_{11} \frac{\partial}{\partial t} (\vec{\nabla} \cdot \vec{u}) N_I \, d\Omega = \sum_{i=1}^{3} \sum_{J=1}^{n} \frac{\partial u_{i_J}}{\partial t} \sum_{K=1}^{n} C_{11_K} \int_{\Omega^e} N_K \frac{\partial N_J}{\partial x_i} N_I \, d\Omega.$$

Where any continuous primary variable $\psi(\Omega, t)$ is a function of time t and space Ω; the value of a primary variable at a node J is $\psi_J(t)$, a function of time only; the trial function $N_I(\Omega)$ a function of space only; $n = 8$ is the number of nodal values; $\vec{u} \equiv u_i = (u_1, u_2, u_3)$ is the displacement vector; $\frac{\partial \vec{u}}{\partial t}$ is the velocity of the solid matrix; and $\vec{n}$ is the normal to the boundary surface Γ.

5.3.4 *Energy equation*

Expansion by term:

$$i)\quad \int_{\Omega^e} D_1 \frac{\partial S_o}{\partial t} N_I \, d\Omega = \sum_{J=1}^{n} \frac{\partial S_{o_J}}{\partial t} \sum_{K=1}^{n} D_{1_K} \int_{\Omega^e} N_K N_J N_I \, d\Omega.$$

$$ii)\quad \int_{\Omega^e} D_2 \frac{\partial p_w}{\partial t} N_I \, d\Omega = \sum_{J=1}^{n} \frac{\partial p_{w_J}}{\partial t} \sum_{K=1}^{n} D_{2_K} \int_{\Omega^e} N_K N_J N_I \, d\Omega.$$

$$iii)\quad \int_{\Omega^e} D_3 \frac{\partial T}{\partial t} N_I \, d\Omega = \sum_{J=1}^{n} \frac{\partial T_J}{\partial t} \sum_{K=1}^{n} D_{3_K} \int_{\Omega^e} N_K N_J N_I \, d\Omega.$$

$$iv)\quad \int_{\Omega^e} D_4 S_w N_I \, d\Omega = \sum_{J=1}^{n} S_{w_J} \sum_{K=1}^{n} D_{4_K} \int_{\Omega^e} N_K N_J N_I \, d\Omega.$$

$$v)\quad \int_{\Omega^e} D_5 S_o N_I \, d\Omega = \sum_{J=1}^{n} S_{o_J} \sum_{K=1}^{n} D_{5_K} \int_{\Omega^e} N_K N_J N_I \, d\Omega.$$

$$vi)\quad \int_{\Omega^e} D_6 p_w N_I \, d\Omega = \sum_{J=1}^{n} p_{w_J} \sum_{K=1}^{n} D_{6_K} \int_{\Omega^e} N_K N_J N_I \, d\Omega.$$

$$vii)\quad \int_{\Omega^e} \vec{D}_7 \cdot \vec{\nabla} S_o W_I \, d\Omega = \sum_{J=1}^{n} S_{o_J} \sum_{K=1}^{n} \vec{D}_{7_K} \cdot \int_{\Omega^e} N_K \vec{\nabla} N_J W_I \, d\Omega.$$

$$viii)\quad \int_{\Omega^e} \vec{D}_8 \cdot \vec{\nabla} p_w W_I \, d\Omega = \sum_{J=1}^{n} p_{w_J} \sum_{K=1}^{n} \vec{D}_{8_K} \cdot \int_{\Omega^e} N_K \vec{\nabla} N_J W_I \, d\Omega.$$

$$ix)\quad \int_{\Omega^e} \vec{D}_9 \cdot \vec{\nabla} T W_I \, d\Omega = \sum_{J=1}^{n} T_J \sum_{K=1}^{n} \vec{D}_{9_K} \cdot \int_{\Omega^e} N_K \vec{\nabla} N_J W_I \, d\Omega.$$

$$\begin{aligned} x)\quad \int_{\Omega^e} \vec{\nabla} \cdot \left(D_{10} \vec{\nabla} T \right) N_I \, d\Omega &= \int_B \left(D_{10} \vec{\nabla} T \right) N_I \cdot \vec{n} \, dB \\ &\quad - \sum_{J=1}^{n} T_J \sum_{K=1}^{n} \vec{\vec{D}}_{10_K} \int_{\Omega^e} N_K \vec{\nabla} N_J \cdot \vec{\nabla} N_I \, d\Omega. \end{aligned}$$

$$xi)\quad \int_{\Omega^e} D_{11} \frac{\partial}{\partial t} (\vec{\nabla} \cdot \vec{u}) N_I \, d\Omega = \sum_{i=1}^{3} \sum_{J=1}^{n} \frac{\partial u_{i_J}}{\partial t} \sum_{K=1}^{n} D_{11_K} \int_{\Omega^e} N_K \frac{\partial N_J}{\partial x_i} N_I \, d\Omega.$$

$$xii)\quad \int_{\Omega^e} \vec{D}_{12} \cdot \frac{\partial \vec{u}}{\partial t} N_I \, d\Omega = \sum_{J=1}^{n} \frac{\partial u_{3_J}}{\partial t} \sum_{K=1}^{n} D_{12_{(3)K}} \int_{\Omega^e} N_K N_J N_I \, d\Omega.$$

$$xiii)\quad \int_{\Omega^e} D_{13_{NC}} N_I \, d\Omega = \sum_{J=1}^{n} D_{13_{NC_J}} \int_{\Omega^e} N_J N_I \, d\Omega.$$

$$xiv)\quad \int_{\Omega^e} D_{13_C} W_I \, d\Omega = \sum_{J=1}^{n} D_{13_{C_J}} \int_{\Omega^e} N_J W_I \, d\Omega.$$

Where any continuous primary variable $\psi(\Omega, t)$ is a function of time t and space Ω; the value of a primary variable at a node J is $\psi_J(t)$, a function of time only; the trial function $N_I(\Omega)$ a function of space only; $n = 8$ is the number of nodal values; $\vec{u} \equiv u_i = (u_1, u_2, u_3)$ is the displacement vector; $\frac{\partial \vec{u}}{\partial t}$ is the velocity of the solid matrix; and $\vec{n}$ is the normal to the boundary surface Γ.

5.4.1 *Row 1: Water equation*

Coefficients for each term in this equation:

$$
\begin{aligned}
S_{w_J}: &\quad \left[S^{11}_{IJ}\right]^T = A_{5_K}\int_{\Omega^e} N_K N_J N_I \, d\Omega. \\
S_{o_J}: &\quad \left[S^{12}_{IJ}\right]^T = A_{6_K}\int_{\Omega^e} N_K N_J N_I \, d\Omega. \\
p_{w_J}: &\quad \left[S^{13}_{IJ}\right]^T = -\vec{\bar{A}}_{7_K} : \int_{\Omega^e} N_K \vec{\nabla} N_J \vec{\nabla} W_I \, d\Omega. \\
T_J: &\quad \left[S^{14}_{IJ}\right]^T = 0. \\
u_{1_J}: &\quad \left[S^{15}_{IJ}\right]^T = 0. \\
u_{2_J}: &\quad \left[S^{16}_{IJ}\right]^T = 0. \\
u_{3_J}: &\quad \left[S^{17}_{IJ}\right]^T = 0.
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial S_{w_J}}{\partial t}: &\quad \left[D^{11}_{IJ}\right]^T = A_{1_K}\int_{\Omega^e} N_K N_J N_I \, d\Omega. \\
\frac{\partial S_{o_J}}{\partial t}: &\quad \left[D^{12}_{IJ}\right]^T = A_{2_K}\int_{\Omega^e} N_K N_J N_I \, d\Omega. \\
\frac{\partial p_{w_J}}{\partial t}: &\quad \left[D^{13}_{IJ}\right]^T = A_{3_K}\int_{\Omega^e} N_K N_J N_I \, d\Omega. \\
\frac{\partial T_J}{\partial t}: &\quad \left[D^{14}_{IJ}\right]^T = A_{4_K}\int_{\Omega^e} N_K N_J N_I \, d\Omega. \\
\frac{\partial u_{1_J}}{\partial t}: &\quad \left[D^{15}_{IJ}\right]^T = A_{10_K}\int_{\Omega^e} N_K \frac{\partial N_J}{\partial x_1} N_I \, d\Omega. \\
\frac{\partial u_{2_J}}{\partial t}: &\quad \left[D^{16}_{IJ}\right]^T = A_{10_K}\int_{\Omega^e} N_K \frac{\partial N_J}{\partial x_2} N_I \, d\Omega. \\
\frac{\partial u_{3_J}}{\partial t}: &\quad \left[D^{17}_{IJ}\right]^T = A_{10_K}\int_{\Omega^e} N_K \frac{\partial N_J}{\partial x_3} N_I \, d\Omega.
\end{aligned}
$$

$$
F_I: \quad F^1_I = \int_\Gamma \left(\vec{\bar{A}}_7 \cdot \vec{\nabla} p_w\right) W_I \cdot \vec{n} \, d\Gamma + \vec{A}_{8_J} \cdot \int_{\Omega^e} \vec{\nabla} N_J W_I \, d\Omega + A_{9_J}\int_{\Omega^e} N_J N_I \, d\Omega.
$$

5.4 *The element equation*

The spatially-discretized element equation can be written as an agglomeration of the physical equations in matrix form as follows, where the superscript α is the equation type (water, bitumen, gas, energy), and β indicates to which variable the 8 × 3 coefficient submatrix belongs:

$$\mathfrak{M}_I^\alpha = S_{IJ}^{\alpha\beta}\psi_J^\beta + D_{IJ}^{\alpha\beta}\frac{\partial \psi_J^\beta}{\partial t} + F_I^\alpha = 0. \tag{5.16}$$

This equation is in indicial notation as described at the start of this section, so that it can be expanded as

$$\begin{Bmatrix} \mathfrak{M}_I^1 \\ \mathfrak{M}_I^2 \\ \mathfrak{M}_I^3 \\ \mathfrak{M}_I^4 \\ \mathfrak{M}_I^5 \\ \mathfrak{M}_I^6 \\ \mathfrak{M}_I^7 \end{Bmatrix} = \begin{bmatrix} S_{IJ}^{11} & S_{IJ}^{12} & S_{IJ}^{13} & S_{IJ}^{14} & S_{IJ}^{15} & S_{IJ}^{16} & S_{IJ}^{17} \\ S_{IJ}^{21} & S_{IJ}^{22} & S_{IJ}^{23} & S_{IJ}^{24} & S_{IJ}^{25} & S_{IJ}^{26} & S_{IJ}^{27} \\ S_{IJ}^{31} & S_{IJ}^{32} & S_{IJ}^{33} & S_{IJ}^{34} & S_{IJ}^{35} & S_{IJ}^{36} & S_{IJ}^{37} \\ S_{IJ}^{41} & S_{IJ}^{42} & S_{IJ}^{43} & S_{IJ}^{44} & S_{IJ}^{45} & S_{IJ}^{46} & S_{IJ}^{47} \\ S_{IJ}^{51} & S_{IJ}^{52} & S_{IJ}^{53} & S_{IJ}^{54} & S_{IJ}^{55} & S_{IJ}^{56} & S_{IJ}^{57} \\ S_{IJ}^{61} & S_{IJ}^{62} & S_{IJ}^{63} & S_{IJ}^{64} & S_{IJ}^{65} & S_{IJ}^{66} & S_{IJ}^{67} \\ S_{IJ}^{71} & S_{IJ}^{72} & S_{IJ}^{73} & S_{IJ}^{74} & S_{IJ}^{75} & S_{IJ}^{76} & S_{IJ}^{77} \end{bmatrix} \begin{Bmatrix} S_{w_J} \\ S_{o_J} \\ p_{w_J} \\ T_J \\ u_{1_J} \\ u_{2_J} \\ u_{3_J} \end{Bmatrix} +$$

$$\begin{bmatrix} D_{IJ}^{11} & D_{IJ}^{12} & D_{IJ}^{13} & D_{IJ}^{14} & D_{IJ}^{15} & D_{IJ}^{16} & D_{IJ}^{17} \\ D_{IJ}^{21} & D_{IJ}^{22} & D_{IJ}^{23} & D_{IJ}^{24} & D_{IJ}^{25} & D_{IJ}^{26} & D_{IJ}^{27} \\ D_{IJ}^{31} & D_{IJ}^{32} & D_{IJ}^{33} & D_{IJ}^{34} & D_{IJ}^{35} & D_{IJ}^{36} & D_{IJ}^{37} \\ D_{IJ}^{41} & D_{IJ}^{42} & D_{IJ}^{43} & D_{IJ}^{44} & D_{IJ}^{45} & D_{IJ}^{46} & D_{IJ}^{47} \\ D_{IJ}^{51} & D_{IJ}^{52} & D_{IJ}^{53} & D_{IJ}^{54} & D_{IJ}^{55} & D_{IJ}^{56} & D_{IJ}^{57} \\ D_{IJ}^{61} & D_{IJ}^{62} & D_{IJ}^{63} & D_{IJ}^{64} & D_{IJ}^{65} & D_{IJ}^{66} & D_{IJ}^{67} \\ D_{IJ}^{71} & D_{IJ}^{72} & D_{IJ}^{73} & D_{IJ}^{74} & D_{IJ}^{75} & D_{IJ}^{76} & D_{IJ}^{77} \end{bmatrix} \begin{Bmatrix} \frac{\partial S_{w_J}}{\partial t} \\ \frac{\partial S_{o_J}}{\partial t} \\ \frac{\partial p_{w_J}}{\partial t} \\ \frac{\partial T_J}{\partial t} \\ \frac{\partial u_{1_J}}{\partial t} \\ \frac{\partial u_{2_J}}{\partial t} \\ \frac{\partial u_{3_J}}{\partial t} \end{Bmatrix} + \begin{Bmatrix} F_I^1 \\ F_I^2 \\ F_I^3 \\ F_I^4 \\ F_I^5 \\ F_I^6 \\ F_I^7 \end{Bmatrix} = 0. \tag{5.17}$$

where $I, J = [1..8]$, the nodal points; thus each entry $S_{IJ}^{\alpha\beta}, D_{IJ}^{\alpha\beta}$ is an 8 × 8 submatrix, while each subvector ψ_J^β, $\frac{\partial \psi_J^\beta}{\partial t}$, F_I^α contains eight elements for each $\alpha, \beta = [1..7]$.

Note that we have moved the ψ_J^β and $\frac{\partial \psi_J^\beta}{\partial t}$ terms to the right hand side of their coefficients. Most of the terms on the previous pages are vector-dyad dot products, whereas Equation (5.17) contains dyad-vector dot products. To make the terms equivalent, the 8 × 8 submatrices must be transposed:

$$\begin{aligned} S_{IJ}^{\alpha\beta}\psi_J^\beta &= \psi_J^\beta \left[S_{IJ}^{\alpha\beta}\right]^T, \\ D_{IJ}^{\alpha\beta}\frac{\partial \psi_J^\beta}{\partial t} &= \frac{\partial \psi_J^\beta}{\partial t}\left[D_{IJ}^{\alpha\beta}\right]^T. \end{aligned} \tag{5.18}$$

The transposed coefficients $S_{IJ}^{\alpha\beta}$ and $D_{IJ}^{\alpha\beta}$ are presented on the following pages.

5.4.1 *Row 1: Water equation*

Coefficients for each term in this equation:

$$S_{w_J}: \quad \left[S^{11}_{IJ}\right]^T = A_{6_K} \int_{\Omega^e} N_K N_J N_I \, d\Omega.$$

$$S_{o_J}: \quad \left[S^{12}_{IJ}\right]^T = A_{6_K} \int_{\Omega^e} N_K N_J N_I \, d\Omega.$$

$$p_{w_J}: \quad \left[S^{13}_{IJ}\right]^T = -\vec{\vec{A}}_{7_K} : \int_{\Omega^e} N_K \vec{\nabla} N_J \vec{\nabla} W_I \, d\Omega.$$

$$T_J: \quad \left[S^{14}_{IJ}\right]^T = 0.$$

$$u_{1_J}: \quad \left[S^{15}_{IJ}\right]^T = 0.$$

$$u_{2_J}: \quad \left[S^{16}_{IJ}\right]^T = 0.$$

$$u_{3_J}: \quad \left[S^{17}_{IJ}\right]^T = 0.$$

$$\frac{\partial S_{w_J}}{\partial t}: \quad \left[D^{11}_{IJ}\right]^T = A_{1_K} \int_{\Omega^e} N_K N_J N_I \, d\Omega.$$

$$\frac{\partial S_{o_J}}{\partial t}: \quad \left[D^{12}_{IJ}\right]^T = A_{2_K} \int_{\Omega^e} N_K N_J N_I \, d\Omega.$$

$$\frac{\partial p_{w_J}}{\partial t}: \quad \left[D^{13}_{IJ}\right]^T = A_{3_K} \int_{\Omega^e} N_K N_J N_I \, d\Omega.$$

$$\frac{\partial T_J}{\partial t}: \quad \left[D^{14}_{IJ}\right]^T = A_{4_K} \int_{\Omega^e} N_K N_J N_I \, d\Omega.$$

$$\frac{\partial u_{1_J}}{\partial t}: \quad \left[D^{15}_{IJ}\right]^T = A_{10_K} \int_{\Omega^e} N_K \frac{\partial N_J}{\partial x_1} N_I \, d\Omega.$$

$$\frac{\partial u_{2_J}}{\partial t}: \quad \left[D^{16}_{IJ}\right]^T = A_{10_K} \int_{\Omega^e} N_K \frac{\partial N_J}{\partial x_2} N_I \, d\Omega.$$

$$\frac{\partial u_{3_J}}{\partial t}: \quad \left[D^{17}_{IJ}\right]^T = A_{10_K} \int_{\Omega^e} N_K \frac{\partial N_J}{\partial x_3} N_I \, d\Omega.$$

$$F_I: \quad F^1_I = \int_{\Gamma} \left(\vec{\vec{A}}_7 \cdot \vec{\nabla} p_w \right) W_I \cdot \vec{n} \, d\Gamma + \vec{A}_{8_J} \cdot \int_{\Omega^e} \vec{\nabla} N_J W_I \, d\Omega + A_{9_J} \int_{\Omega^e} N_J N_I \, d\Omega.$$

5.4.2 *Row 2: Bitumen (oil) equation*

Coefficients for each term in this equation:

$$
\begin{aligned}
S_{w_J} :&\quad \left[S^{21}_{IJ}\right]^T = 0.\\
S_{o_J} :&\quad \left[S^{22}_{IJ}\right]^T = B_{4_K}\int_{\Omega^e} N_K N_J N_I \, d\Omega - \vec{\vec{B}}_{6_K} : \int_{\Omega^e} N_K \vec{\nabla} N_J \vec{\nabla} W_I \, d\Omega.\\
p_{w_J} :&\quad \left[S^{23}_{IJ}\right]^T = -\vec{\vec{B}}_{5_K} : \int_{\Omega^e} N_K \vec{\nabla} N_J \vec{\nabla} W_I \, d\Omega.\\
T_J :&\quad \left[S^{24}_{IJ}\right]^T = 0.\\
u_{1_J} :&\quad \left[S^{25}_{IJ}\right]^T = 0.\\
u_{2_J} :&\quad \left[S^{26}_{IJ}\right]^T = 0.\\
u_{3_J} :&\quad \left[S^{27}_{IJ}\right]^T = 0.
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial S_{w_J}}{\partial t} :&\quad \left[D^{21}_{IJ}\right]^T = 0.\\
\frac{\partial S_{o_J}}{\partial t} :&\quad \left[D^{22}_{IJ}\right]^T = B_{1_K}\int_{\Omega^e} N_K N_J N_I \, d\Omega.\\
\frac{\partial p_{w_J}}{\partial t} :&\quad \left[D^{23}_{IJ}\right]^T = B_{2_K}\int_{\Omega^e} N_K N_J N_I \, d\Omega.\\
\frac{\partial T_J}{\partial t} :&\quad \left[D^{24}_{IJ}\right]^T = B_{3_K}\int_{\Omega^e} N_K N_J N_I \, d\Omega.\\
\frac{\partial u_{1_J}}{\partial t} :&\quad \left[D^{25}_{IJ}\right]^T = B_{9_K}\int_{\Omega^e} N_K \frac{\partial N_J}{\partial x_1} N_I \, d\Omega.\\
\frac{\partial u_{2_J}}{\partial t} :&\quad \left[D^{26}_{IJ}\right]^T = B_{9_K}\int_{\Omega^e} N_K \frac{\partial N_J}{\partial x_2} N_I \, d\Omega.\\
\frac{\partial u_{3_J}}{\partial t} :&\quad \left[D^{27}_{IJ}\right]^T = B_{9_K}\int_{\Omega^e} N_K \frac{\partial N_J}{\partial x_3} N_I \, d\Omega.
\end{aligned}
$$

$$
\begin{aligned}
F_I :\quad F^2_I = &\int_\Gamma \left(\vec{\vec{B}}_5 \cdot \vec{\nabla} p_w\right) W_I \cdot \vec{n} \, d\Gamma + \int_\Gamma \left(\vec{\vec{B}}_6 \cdot \vec{\nabla} S_o\right) W_I \cdot \vec{n} \, d\Gamma\\
&+ \vec{B}_{7_J} \cdot \int_{\Omega^e} \vec{\nabla} N_J W_I \, d\Omega + B_{8_J}\int_{\Omega^e} N_J N_I \, d\Omega.
\end{aligned}
$$

5.4.3 *Row 3: Gas equation*

Coefficients for each term in this equation:

$$S_{w_J}: \quad \left[S^{31}_{IJ}\right]^T = C_{5_K} \int_{\Omega^e} N_K N_J N_I \, d\Omega.$$

$$S_{o_J}: \quad \left[S^{32}_{IJ}\right]^T = C_{6_K} \int_{\Omega^e} N_K N_J N_I \, d\Omega - \vec{\vec{C}}_{8_K} : \int_{\Omega^e} N_K \vec{\nabla} N_J \vec{\nabla} W_I \, d\Omega.$$

$$p_{w_J}: \quad \left[S^{33}_{IJ}\right]^T = -\vec{\vec{C}}_{7_K} : \int_{\Omega^e} N_K \vec{\nabla} N_J \vec{\nabla} W_I \, d\Omega.$$

$$T_J: \quad \left[S^{34}_{IJ}\right]^T = 0.$$

$$u_{1_J}: \quad \left[S^{35}_{IJ}\right]^T = 0.$$

$$u_{2_J}: \quad \left[S^{36}_{IJ}\right]^T = 0.$$

$$u_{3_J}: \quad \left[S^{37}_{IJ}\right]^T = 0.$$

$$\frac{\partial S_{w_J}}{\partial t}: \quad \left[D^{31}_{IJ}\right]^T = C_{1_K} \int_{\Omega^e} N_K N_J N_I \, d\Omega.$$

$$\frac{\partial S_{o_J}}{\partial t}: \quad \left[D^{32}_{IJ}\right]^T = C_{2_K} \int_{\Omega^e} N_K N_J N_I \, d\Omega.$$

$$\frac{\partial p_{w_J}}{\partial t}: \quad \left[D^{33}_{IJ}\right]^T = C_{3_K} \int_{\Omega^e} N_K N_J N_I \, d\Omega.$$

$$\frac{\partial T_J}{\partial t}: \quad \left[D^{34}_{IJ}\right]^T = C_{4_K} \int_{\Omega^e} N_K N_J N_I \, d\Omega.$$

$$\frac{\partial u_{1_J}}{\partial t}: \quad \left[D^{35}_{IJ}\right]^T = C_{11_K} \int_{\Omega^e} N_K \frac{\partial N_J}{\partial x_1} N_I \, d\Omega.$$

$$\frac{\partial u_{2_J}}{\partial t}: \quad \left[D^{36}_{IJ}\right]^T = C_{11_K} \int_{\Omega^e} N_K \frac{\partial N_J}{\partial x_2} N_I \, d\Omega.$$

$$\frac{\partial u_{3_J}}{\partial t}: \quad \left[D^{37}_{IJ}\right]^T = C_{11_K} \int_{\Omega^e} N_K \frac{\partial N_J}{\partial x_3} N_I \, d\Omega.$$

$$F_I: \quad F^3_I = \int_\Gamma \left(\vec{\vec{C}}_7 \cdot \vec{\nabla} p_w\right) W_I \cdot \vec{n} \, d\Gamma + \int_\Gamma \left(\vec{\vec{C}}_8 \cdot \vec{\nabla} S_o\right) W_I \cdot \vec{n} \, d\Gamma$$

$$+\vec{C}_{9_J} \cdot \int_{\Omega^e} \vec{\nabla} N_J W_I \, d\Omega + C_{10_J} \int_{\Omega^e} N_J N_I \, d\Omega.$$

5.4.4 *Row 4: Energy equation*

Coefficients for each term in this equation:

$$S_{w_J}: \quad \left[S_{IJ}^{41}\right]^T = D_{4_K} \int_{\Omega^e} N_K N_J N_I \, d\Omega.$$

$$S_{o_J}: \quad \left[S_{IJ}^{42}\right]^T = D_{5_K} \int_{\Omega^e} N_K N_J N_I \, d\Omega + \vec{D}_{7_K} \cdot \int_{\Omega^e} N_K \vec{\nabla} N_J W_I \, d\Omega.$$

$$p_{w_J}: \quad \left[S_{IJ}^{43}\right]^T = D_{6_K} \int_{\Omega^e} N_K N_J N_I \, d\Omega + \vec{D}_{8_K} \cdot \int_{\Omega^e} N_K \vec{\nabla} N_J W_I \, d\Omega.$$

$$T_J: \quad \left[S_{IJ}^{44}\right]^T = \vec{D}_{9_K} \cdot \int_{\Omega^e} N_K \vec{\nabla} N_J W_I \, d\Omega - D_{10_K} \int_{\Omega^e} N_K \vec{\nabla} N_J \cdot \vec{\nabla} N_I \, d\Omega.$$

$$u_{1_J}: \quad \left[S_{IJ}^{45}\right]^T = 0.$$

$$u_{2_J}: \quad \left[S_{IJ}^{46}\right]^T = 0.$$

$$u_{3_J}: \quad \left[S_{IJ}^{47}\right]^T = 0.$$

$$\frac{\partial S_{w_J}}{\partial t}: \quad \left[D_{IJ}^{41}\right]^T = 0.$$

$$\frac{\partial S_{o_J}}{\partial t}: \quad \left[D_{IJ}^{42}\right]^T = D_{1_K} \int_{\Omega^e} N_K N_J N_I \, d\Omega.$$

$$\frac{\partial p_{w_J}}{\partial t}: \quad \left[D_{IJ}^{43}\right]^T = D_{2_K} \int_{\Omega^e} N_K N_J N_I \, d\Omega.$$

$$\frac{\partial T_J}{\partial t}: \quad \left[D_{IJ}^{44}\right]^T = D_{3_K} \int_{\Omega^e} N_K N_J N_I \, d\Omega.$$

$$\frac{\partial u_{1_J}}{\partial t}: \quad \left[D_{IJ}^{45}\right]^T = D_{11_K} \int_{\Omega^e} N_K \frac{\partial N_J}{\partial x_1} N_I \, d\Omega.$$

$$\frac{\partial u_{2_J}}{\partial t}: \quad \left[D_{IJ}^{46}\right]^T = D_{11_K} \int_{\Omega^e} N_K \frac{\partial N_J}{\partial x_2} N_I \, d\Omega.$$

$$\frac{\partial u_{3_J}}{\partial t}: \quad \left[D_{IJ}^{47}\right]^T = D_{11_K} \int_{\Omega^e} N_K \frac{\partial N_J}{\partial x_3} N_I \, d\Omega + D_{12_{(3)K}} \int_{\Omega^e} N_K N_J N_I \, d\Omega.$$

$$F_I: \quad F_I^4 = \int_\Gamma \left(D_9 \vec{\nabla} T\right) N_I \cdot \vec{n} \, d\Gamma + D_{13_{NC_J}} \int_{\Omega^e} N_J N_I \, d\Omega + D_{13_{C_J}} \int_{\Omega^e} N_J W_I \, d\Omega.$$

5.4.5 *Row 5:* x*-direction displacement*

Coefficients for each term in this equation:

S_{w_J} : $S^{51}_{IJ} = 0.$

S_{o_J} : $S^{52}_{IJ} = 0.$

p_{w_J} : $S^{53}_{IJ} = -\int_{\Omega^e} \frac{\partial N_I}{\partial x_1} N_J \, d\Omega.$

T_J : $S^{54}_{IJ} = -\int_{\Omega^e} [\alpha(2G + 3\lambda)] \frac{\partial N_I}{\partial x_1} N_J \, d\Omega.$

u_{1_J} : S^{55}_{IJ}: see the structural stiffness matrix.

u_{2_J} : S^{56}_{IJ}: see the structural stiffness matrix.

u_{3_J} : S^{57}_{IJ}: see the structural stiffness matrix.

$$\text{All} \quad D^{5\beta}_{IJ} = 0, \qquad \beta = [1..7].$$

F_I : $F^5_I = \int_{\Omega^e} \frac{\partial N_I}{\partial x_j} \sigma^{\circ}_{1j} \, d\Omega - \int_{\Omega^e} N_I F_1 \, d\Omega - \int_{\Gamma^e_2} N_I S_1 \, d\Gamma.$

Note that on the surface traction boundary

$$S_1 = \tau_{1j} n_j.$$

Only the body force component in the z-direction, F_3, is non-zero. This is the gravitational acceleration vector.

5.4.6 *Row 6:* **y**-*direction displacement*

Coefficients for each term in this equation:

S_{w_J} : $S^{61}_{IJ} = 0.$

S_{o_J} : $S^{62}_{IJ} = 0.$

p_{w_J} : $S^{63}_{IJ} = -\int_{\Omega^*} \frac{\partial N_I}{\partial x_2} N_J \, d\Omega.$

T_J : $S^{64}_{IJ} = -\int_{\Omega^*} [\alpha(2G + 3\lambda)] \frac{\partial N_I}{\partial x_2} N_J \, d\Omega.$

u_{1_J} : S^{65}_{IJ}: see the structural stiffness matrix.

u_{2_J} : S^{66}_{IJ}: see the structural stiffness matrix.

u_{3_J} : S^{67}_{IJ}: see the structural stiffness matrix.

$$\text{All} \quad D^{6\beta}_{IJ} = 0, \qquad \beta = [1..7].$$

$$F_I : \quad F^6_I = \int_{\Omega^*} \frac{\partial N_I}{\partial x_j} \sigma^\circ_{2j} \, d\Omega - \int_{\Omega^*} N_I F_2 \, d\Omega - \int_{\Gamma^*_2} N_I S_2 \, d\Gamma.$$

Note that on the surface traction boundary

$$S_2 = \tau_{2j} n_j.$$

Only the body force component in the z-direction, F_3, is non-zero. This is the gravitational acceleration vector.

5.4.7 *Row 7: z-direction displacement*

- Coefficients for each term in this equation:

S_{w_J} : $S^{71}_{IJ} = 0.$

S_{o_J} : $S^{72}_{IJ} = 0.$

p_{w_J} : $S^{73}_{IJ} = -\int_{\Omega^e} \frac{\partial N_I}{\partial x_3} N_J \, d\Omega.$

T_J : $S^{74}_{IJ} = -\int_{\Omega^e} [\alpha(2G + 3\lambda)] \frac{\partial N_I}{\partial x_3} N_J \, d\Omega.$

u_{1_J} : S^{75}_{IJ}: see the structural stiffness matrix.

u_{2_J} : S^{76}_{IJ}: see the structural stiffness matrix.

u_{3_J} : S^{77}_{IJ}: see the structural stiffness matrix.

All $D^{7\beta}_{IJ} = 0, \qquad \beta = [1..7].$

F_I : $F^7_I = \int_{\Omega^e} \frac{\partial N_I}{\partial x_j} \sigma^{\circ}_{3j} \, d\Omega - \int_{\Omega^e} N_I F_3 \, d\Omega - \int_{\Gamma^e_2} N_I S_3 \, d\Gamma.$

Note that on the surface traction boundary

$$S_3 = \tau_{3j} n_j.$$

Only the body force component in the z-direction, F_3, is non-zero. This is the gravitational acceleration vector.

5.5 *Structural stiffness matrix*

The block of submatrices

$$\begin{bmatrix} S_{IJ}^{55} & S_{IJ}^{56} & S_{IJ}^{57} \\ S_{IJ}^{65} & S_{IJ}^{66} & S_{IJ}^{67} \\ S_{IJ}^{75} & S_{IJ}^{76} & S_{IJ}^{77} \end{bmatrix} \tag{5.19}$$

is defined as structural stiffness matrix. This is given by

$$[K_{IJ}]^e = \int_{\Omega^e} [B_I]^T [D] [B_J] \, d\Omega. \tag{5.20}$$

where $[D]$ contains only material dependent parameters—the constitutive elasticity matrix relating stress and strain—and [B] relates strain to nodal displacements. For elastoplastic material behaviour the constitutive matrix is formed from:

$$[D_{ep}] = [D] \left[[I] - \left\{ \frac{\partial G}{\partial \sigma} \right\} \left(\frac{\left\{\frac{\partial F}{\partial \sigma}\right\}^T [D]}{S + \left\{\frac{\partial F}{\partial \sigma}\right\}^T [D] \left\{\frac{\partial G}{\partial \sigma}\right\}} \right) \right]. \tag{5.21}$$

5.6 *A general time integration method for differential equations using finite elements*

Given a spatially integrated general system of equations arising from any non-linear partial differential equation:

$$S_{ij}(\psi, t)\psi_j(t) + D_{ij}(\psi, t)\frac{d\psi_j(t)}{dt} + F_i(\psi, t) = 0. \tag{5.22}$$

where the subscripts i, j vary over the element nodes [1..**nnodes**]. This can be written more simply as:

$$S_{ij}\psi_j + D_{ij}\frac{d\psi_j}{dt} + F_i = 0. \tag{5.22}$$

where ψ_j are nodal values of S_w, S_o, p_w, T, and the displacements u_i in our problem.

We can follow the approach of Zienkiewicz and Morgan (1983) but using non-linear coefficients instead of linear. We approximate the time variable by linear finite elements. This is illustrated in Figure 5–2.

We can then approximate ψ_j by

$$\hat{\psi}_j = \sum_{m=1}^{\infty} \psi_j^m N_m. \tag{5.23}$$

where ψ_j^m is the value of the approximation $\hat{\psi}_j$ at time node m.

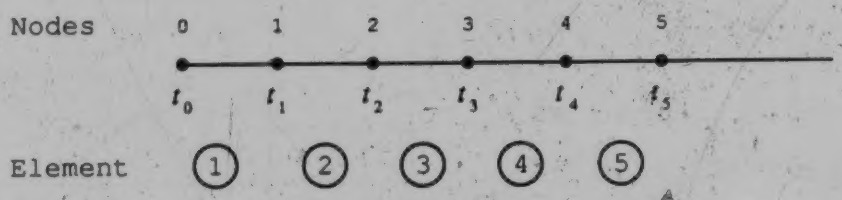

Nodes
0
1
2
3
4
5
t_0
t_1
t_2
t_3
t_4
t_5
Element
1
2
3
4
5

Figure 5–2

Linear finite elements of time.

We use the same shape functions N_m for all elements to represent the variation of each component of ψ_j. N_m is a scalar quantity and we have chosen a linear polynomial. Consider the nth element, surrounded by nodes at $t = t_n$ and $t = t_{n+1}$:

$$\hat{\psi}_j = \psi_j^n N_n^n + \psi_j^{n+1} N_{n+1}^n. \tag{5.24}$$

because all the other trial functions are zero within element n. This can be shown diagrammatically as in Figure 5–3.

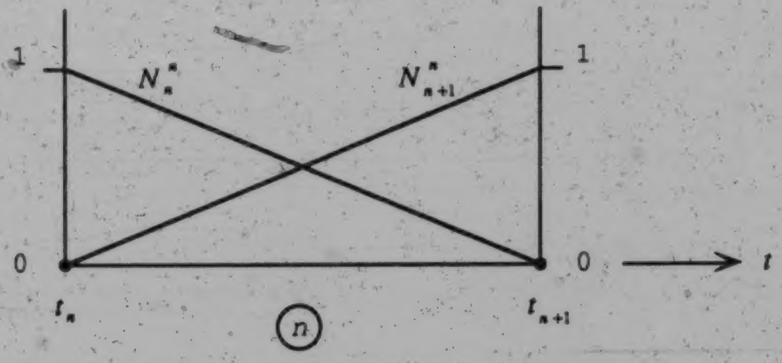

1
N_n^n
N_{n+1}^n
1
0
0
t
t_n
n
t_{n+1}

Figure 5–3

Linear finite elements of time.

Now, the element basis functions are described as:

$$\left.\begin{array}{ll} N_n^n = 1 - T, & \dfrac{dN_n^n}{dt} = \dfrac{-1}{\Delta t_n}; \\[2ex] N_{n+1}^n = T, & \dfrac{dN_{n+1}^n}{dt} = \dfrac{1}{\Delta t_n}. \end{array}\right\} \tag{5.25}$$

where the local coordinate T is defined as

$$T = \frac{t - t_n}{\Delta t_n}, \quad \text{and} \quad \Delta t_n = t_{n+1} - t_n. \tag{5.26}$$

In other words, T varies between 0 and 1 from $t = t_n$ to t_{n+1}.

Apply the technique of weighted residuals over all time elements:

$$\int_0^\infty \left(D_{ij}\frac{d\hat{\psi}_j}{dt} + S_{ij}\hat{\psi}_j + F_i\right) W_n \, dt, \quad \text{where} \quad n = 0, 1, 2, \ldots \tag{5.27}$$

Considering only those weighting functions W_n such that:

$$W_n = 0, \qquad t < t_n \text{ and } t > t_{n+1} \tag{5.28}$$

then the integral becomes

$$\int_0^1 \left(D_{ij}\frac{d\hat{\psi}_j}{dt} + S_{ij}\hat{\psi}_j + F_i\right) W_n \, dt, \quad \text{where} \quad n = 0, 1, 2, \ldots \tag{5.29}$$

This integral extends only over element n. Use Equation (5.24) to replace ψ_j, and use Equation (5.25) to obtain Equation(5.29). (Note that D_{ij}, S_{ij} and F_i are functions of time):

$$\int_0^1 \left\{D_{ij}(t_n + \Delta t_n T)\frac{\Delta\psi_j^n}{\Delta t_n} + S_{ij}(t_n + \Delta t_n T)[\psi_j^n(1-T) + \psi_j^{n+1}T]\right\} W_n \, dT + \int_0^1 [F_i(t_n + \Delta t_n T)] \, W_n \, dT = 0. \tag{5.30}$$

where $\Delta t_n = t_{n+1} - t_n$ and $\Delta\psi_j^n = \psi_j^{n+1} - \psi_j^n$. On expansion, Equation (5.30) becomes:

$$\begin{aligned} &\left[\frac{1}{\Delta t_n}\int_0^1 D_{ij}(t_n + \Delta t_n T) W_n \, dT + \int_0^1 S_{ij}(t_n + \Delta t_n T) T W_n \, dT\right]\psi_j^{n+1} \\ &\quad + \left[\frac{-1}{\Delta t_n}\int_0^1 D_{ij}(t_n + \Delta t_n T) W_n \, dT + \int_0^1 S_{ij}(t_n + \Delta t_n T)(1 - T) W_n \, dT\right]\psi_j^n \\ &\quad + \int_0^1 F_i(t_n + \Delta t_n T) W_n \, dT = 0. \end{aligned} \tag{5.31}$$

We thus have a two-level time stepping scheme. Now, if D_{ij}, S_{ij} and F_i are smooth functions in time, we can use the same interpolation function as that used for ψ_j:

$$\begin{aligned} f(t_n + T\Delta t_n) &\approx f^n N_n^n(T) + f^{n+1} N_{n+1}^n(T) \\ &= f^n(1-T) + f^{n+1}(T). \qquad 0 \leq T \leq 1 \end{aligned} \tag{5.32}$$

where N^n is a trial function within element n. Then Equation (5.30) becomes:

$$\begin{aligned} &\int_0^1 \{[D_{ij}^n(1-T) + D_{ij}^{n+1}T]\frac{\Delta\psi_j^n}{\Delta t_n} + [S_{ij}^n(1-T) + S_{ij}^{n+1}T][\psi_{ij}^n(1-T) + \psi_{ij}^{n+1}T]\} W_n \, dT \\ &\quad + \int_0^1 [F_{ij}^n(1-T) + F_{ij}^{n+1}T] W_n \, dT = 0. \end{aligned} \tag{5.33}$$

The second term in the first integrand is expanded as:

$$\psi_j^n S_{ij}^n(1 - 2T + T^2) + \psi_j^n S_{ij}^{n+1}(T - T^2) + \psi_j^{n+1} S_{ij}^n(T - T^2) + \psi_j^{n+1} S_{ij}^{n+1} T^2$$

. Thus we have:

$$\begin{aligned} &\frac{\Delta\psi_j^n}{\Delta t_n}\left(D_{ij}^n \int_0^1 W_n \, dT + \Delta D_{ij}^n \int_0^1 T W_n \, dT\right) + S_{ij}^n \Delta\psi_j^n \int_0^1 W_n \, dT \\ &\quad + (S_{ij}^{n+1}\psi_j^n + S_{ij}^n \psi_j^{n+1} - 2S_{ij}^n \psi_j^n) \int_0^1 T W_n \, dT \\ &\quad + (S_{ij}^n \psi_j^n - S_{ij}^{n+1}\psi_j^n - S_{ij}^n \psi_j^{n+1} + S_{ij}^{n+1}\psi_j^{n+1}) \int_0^1 T^2 W_n \, dT \\ &\quad + F_i \int_0^1 W_n \, dT + \Delta F_{ij}^n \int_0^1 T W_n \, dT = 0. \end{aligned} \tag{5.34}$$

Dividing through by $\int_0^1 W_n \, dt$ and noting that $\Delta S^n = S^{n+1} - S^n$, and similarly for the other terms, Equation (5.34) becomes:

$$(D_{ij}^n + \gamma_n \Delta D_{ij}^n)\frac{\Delta\psi_j^n}{\Delta t_n} + S_{ij}^n \psi_j^n + \gamma_n(S_{ij}^n \Delta\psi_j^n + \Delta S_{ij}^n \psi_j^n) + \Lambda_n \Delta S_{ij}^n \Delta\psi_j^n + F_i^n + \gamma_n \Delta F_i^n = 0. \tag{5.35}$$

where

$$\gamma_n = \frac{\int_0^1 T W_n \, dT}{\int_0^1 W_n \, dT} \quad \text{and} \quad \Lambda_n = \frac{\int_0^1 T^2 W_n \, dT}{\int_0^1 W_n \, dT}.$$

We now take point collocation at $T = \theta$ for each element, for $0 \leq \theta \leq 1$. That means taking $W_n = \delta(T - \theta)$ for $n = 0, 1, 2, \ldots$, where $\delta(x - \theta)$ is the Dirac delta function with the properties:

$$\delta(x - \theta)\begin{cases} 0, & \text{if } x \neq \theta; \\ \infty, & \text{if } x = \theta. \end{cases}$$

$$\int_{x<\theta}^{x>\theta} G(x)\delta(x - \theta)\,dx = G(\theta).$$

We shall also use for $0 \leq \theta \leq 1$:

$$\int_0^1 W_n\,dT = 1;$$
$$\int_0^1 TW_n\,dT = \theta;$$
$$\int_0^1 T^2 W_n\,dT = \theta^2.$$

to obtain:

$$[\theta D_{ij}^n + (1-\theta)D_{ij}^n]\frac{\Delta\psi_j^n}{\Delta t_n} + S_{ij}^n\psi_j^n + \theta(\Delta S_{ij}^n\psi_j^n + S_{ij}^n\Delta\psi_j^n) + \theta^2\Delta S_{ij}^n\Delta\psi_j^n + [\theta F_i^n + (1-\theta)F_i^n] = 0. \quad (5.36)$$

This can be simplified to form the final time discretized equation below:

$$(D_{ij}^n + \theta\Delta D_{ij}^n)\frac{\Delta\psi_j^n}{\Delta t_n} + (S_{ij}^n + \theta\Delta S_{ij}^n)(\psi_j^n + \theta\Delta\psi_j^n) + (F_i^n + \theta\Delta F_i^n) = 0. \quad (5.37)$$

where $\Delta\Theta^n = \Theta^{n+1} - \Theta^n$.

5.6.1 *Obtaining finite difference approximations from the time discretized equation*

If we choose suitable collocation points within the time element, the following finite difference schemes are duplicated:

$\theta = 0$: forward difference (Euler) scheme;
$\theta = 1/2$: central difference (Crank-Nicholson) scheme;
$\theta = 1$: backward difference scheme.

If $\theta = 2/3$, an approximation to the Galerkin solution is obtained. The Galerkin condition is

$$W_n = N_{n+1}^n$$
$$= T.$$

Then γ_n and Λ_n are

$$\gamma_n = \frac{\int_0^1 TT\,dT}{\int_0^1 T\,dT} = \frac{1/3}{1/2} = \frac{2}{3};$$
$$\Lambda_n = \frac{\int_0^1 T^2T\,dT}{\int_0^1 T\,dT} = \frac{1/4}{1/2} = \frac{1}{2} \qquad (\neq \gamma_n^2).$$

It is interesting to note that, in the linear problem, the choice of the collocation point $\theta = \frac{2}{3}$ leads exactly to the Galerkin solution for the linear problem.

5.6.2 *Application of the time discretized equation*

Recall the time discretized equation:

$$(S_{ij}^{\alpha\beta^n} + \theta\Delta S_{ij}^{\alpha\beta^n})(\psi_j^{\beta^n} + \theta\Delta\psi_j^{\beta^n}) + (D_{ij}^{\alpha\beta^n} + \theta\Delta D_{ij}^{\alpha\beta^n})\frac{\Delta\psi_j^{\beta^n}}{\Delta t_n} + (F_i^{\alpha^n} + \theta\Delta F_i^{\alpha^n}).$$

where $\Delta S^n = S^{n+1} - S^n$. Let this equation be represented by

$$\mathfrak{M}_i^\alpha = 0. \tag{5.38}$$

where $\mathfrak{M}$ is evaluated at node i, ψ are the primary variables, and α and β cycle through the equation (or variable) numbers: from 1 to 4 or from 1 to 7. Then the Newton-Kantorovich method can be applied to linearize Equation (5.38):

$$\frac{\partial\mathfrak{M}_i^\alpha}{\partial\psi_h^\gamma}\Delta\psi_h^\gamma = -\mathfrak{M}_i^\alpha. \tag{5.39}$$

where the derivative is taken with respect to ψ_h^γ, a primary variable of type γ and at node number h.

5.6.3 *Comparison with Voss's total integration method (TIM)*

To evaluate the derivatives numerically, only four shifts have to be made, in addition to the evaluation of the equations at the new iteration level. Therefore five evaluations of the properties and other variables must be made. In Voss's TIM, a chord slope method, nine evaluations are required in addition to the value at the start of the time step, although iteration convergence is improved if the correct chord is chosen. However, there is no guarantee that the chord chosen does speed up convergence, and there is always the possibility of a flow reversal. This would be a change in the direction of saturation or pressure change which could ruin the convergence. Convergence in

Newton's method is more tolerant of flow reversals over the time step. Nevertheless, the algorithm must still incorporate a check for flow direction after each Newton-Raphson iteration.

5.7 *Application of the Newton-Raphson method*

The general equation discretized in space and time is given for one finite element by

$$\begin{aligned}\mathfrak{M}_I^\alpha = (S_{IJ}^{\alpha\beta^n} + \theta\Delta S_{IJ}^{\alpha\beta^n})(\psi_J^{\beta^n} + \theta\Delta\psi_J^{\beta^n}) + (D_{IJ}^{\alpha\beta^n} + \theta\Delta D_{IJ}^{\alpha\beta^n})\frac{\Delta\psi_J^{\beta^n}}{\Delta t_n} + (F_I^{\alpha^n} + \theta\Delta F_I^{\alpha^n})\\ = 0\end{aligned} \tag{5.40}$$

where $\Delta S^n = S^{n+1} - S^n$, and similarly for the other terms. Time level n is the old time level and time level $(n+1)$ is the new one. $\mathfrak{M}_I$ is the equation evaluated at nodal point I, ψ are the primary variables, and α and β cycle through the equation (or variable) numbers: from 1 to 4 or from 1 to 7.

Equation (5.40) can be written in a shorter form:

$$\mathfrak{M}_I^\alpha = S_{IJ}^{\alpha\beta'}\psi_J^{\beta'} + D_{IJ}^{\alpha\beta'}\frac{\Delta\psi_J^{\beta^n}}{\Delta t_n} + F_I^{\alpha'}. \tag{5.41}$$

where

$$\begin{aligned}S_{IJ}^{\alpha\beta'} &= S_{IJ}^{\alpha\beta^n} + \theta\Delta S_{IJ}^{\alpha\beta^n}\\ &= S_{IJ}^{\alpha\beta^n} + \theta(S_{IJ}^{\alpha\beta^{(n+1)}} - S_{IJ}^{\alpha\beta^n})\\ &= (1-\theta)S_{IJ}^{\alpha\beta^n} + \theta S_{IJ}^{\alpha\beta^{(n+1)}}.\end{aligned} \tag{5.42}$$

and similarly for the other terms.

Equation (5.2) is non-linear due to the dependence of the coefficients at the $(n+1)$ time level on the primary variables. The equations can be linearized by application of a suitable method, and the Newton-Raphson method is a common and suitable method in this case. The method was originally designed for functions of a single variable but can be successfully extended to functions of many independent variables. This particular application is also known as the Newton-Kantorovich method (Bronstein and Semendyayev, 1985). Application of the Newton-Raphson method to our problem gives rise to the following:

$$\frac{\partial\mathfrak{M}_I^\alpha}{\partial\psi_H^\gamma}\Delta\psi_H^\gamma = -\mathfrak{M}_I^\alpha. \tag{5.43}$$

where the derivative is taken with respect to ψ_H^γ, a primary variable of type γ and at node number H. The derivative can be formed numerically by forward-differencing:

$$\frac{\mathfrak{M}_I^\alpha(\psi_H^\gamma + \Delta\psi_H^\gamma) - \mathfrak{M}_I^\alpha(\psi_H^\gamma)}{\Delta\psi_H^\gamma}. \tag{5.44}$$

where the amount $\Delta\psi_H^\gamma$ is known as the shift, and $\mathfrak{M}_I^\alpha(\psi_H^\gamma + \Delta\psi_H^\gamma)$ is known as the shifted variable.

The two non-linear products $S_{IJ}^{\alpha\beta'}\psi_J^{\beta'}$ and $(D_{IJ}^{\alpha\beta'}\Delta\psi_J^{\beta^n})/\Delta t_n$ occurring on the right-hand side of Equation (5.2) require further study for the purpose of forming derivatives. Purely n-level terms are constant so, if identified, they can be discarded when forming the derivatives, perhaps saving some arithmetic. Each of the terms could be expanded into five simpler terms at time level $(n+1)$. It turns out that approximately three times as much computational effort is required to form the derivative from the expansions than from the original terms. However, the $F_I^{\alpha^n}$ terms can be dropped when forming the derivative. Thus we can write

$$\frac{\partial\mathfrak{M}_I^\alpha}{\partial\psi_H^\gamma} = \frac{\partial\mathfrak{M}_I^{\alpha'}}{\partial\psi_H^\gamma}. \tag{5.45}$$

where

$$\mathfrak{M}_I^{\alpha'} = S_{IJ}^{\alpha\beta'}\psi_J^{\beta'} + D_{IJ}^{\alpha\beta'}\frac{\Delta\psi_J^{\beta^n}}{\Delta t_n} + \theta F_I^{\alpha(n+1)}. \tag{5.46}$$

Although full expansion of the non-linear products does not result in saving arithmetic, selective expansion is desirable. Inspection of the definitions of the coefficients S, D, F also reveals work savings. This is elaborated in the following section.

5.8 *Algebraic reduction of the derivatives*

Two methods are available for reducing the amount of work to form the derivatives in the Newton-Raphson Jacobian matrix. One is the reduction of the non-linear products by partially evaluating the derivatives analytically, the other is by understanding the definitions of the coefficients S, D, F. These methods can be combined to save a substantial amount of computational effort.

5.8.1 *Derivatives of the coefficients* S, D, F

Consider a typical coefficient at time level $(n+1)$, S, say, for $\alpha = 1$, $\beta = 1$:

$$S_{IJ}^{11} \doteq \sum_{K=1}^{n} A_{5_K}\int_{\Omega^e} N_K\,[N_J N_I]^T\,d\Omega. \qquad I, J = [1..n]. \tag{5.47}$$

The integral in this equation is independent of the variables ψ^γ_H, $\gamma \neq 5, 6, 7$ (the displacements). The dependence on the displacements can be ignored for the moment. The presence of the multiplier A_{5_K} is significant because it is by definition only affected by events at node K. Therefore, on taking the derivative of this expression with respect to a variable at node H, we only have a contribution to the derivative from node $K = H$. Thus

$$\frac{\partial A_{5_K}}{\partial \psi^7_H} \equiv \frac{\partial (A_{5_K})}{\partial \psi^7_H}\delta_{KH}. \tag{5.48}$$

where δ_{KH} is the Kronecker delta, so that

$$\frac{\partial S^{11}_{IJ}}{\partial \psi^7_H} = \frac{\partial (A_{5_K})}{\partial \psi^7_H}\delta_{KH}\int_{\Omega^e} N_K \left[N_J N_I\right]^T \, d\Omega. \qquad I, J = [1..n]. \tag{5.49}$$

In other words, the right-hand-side expression in Equation (5.47) need only be evaluated for $K = H$, and there is only one term in the summation for each value of H. This is an important conclusion not merely because we do not need to sum over all $K = [1..n]$. It also permits us to evaluate derivatives at all the nodes by shifting a primary variable at all the nodes at the same time. If $A_{5_{K=H}}$ was a function of the degree of freedom at other nodes $K \neq H$, then each node would have to be shifted in turn to evaluate a derivative with respect to ψ^7_H. As all the properties must be re-evaluated after each shift, there is consequently a factor of eight saving in the amount of work to be done in finding the properties and determining the derivatives. The saving is therefore very substantial.

Note that the definitions of $D^{\alpha\beta}_{IJ}$ and F^α_I are very similar to that of $S^{\alpha\beta}_{IJ}$, and that the derivatives will be formed similarly also. In particular, for F^α_I:

$$\frac{\partial F^\alpha_I}{\partial \psi^7_H} = \frac{\partial (a_J)}{\partial \psi^7_H}\delta_{JH}\int_{\Omega^e} \left[N_J N_I\right]^T \, d\Omega. \qquad I = [1..n]. \tag{5.50}$$

5.8.2 *Derivatives of the non-linear products*

It was discussed earlier that a full expansion of the non-linear products to eliminate purely n-level terms was ineffective. Further inspection of the products shows that the terms can be reduced to forms requiring less work to evaluate numerically. The two non-linear products, in the short notation, are

$$S^{\alpha\beta'}_{IJ}\psi^{\beta'}_J \qquad \text{and} \qquad D^{\alpha\beta'}_{IJ}\frac{\Delta\psi^{\beta^n}_J}{\Delta t_n}. \tag{5.51}$$

Starting with the first term, we can partially evaluate the derivatives using the usual rules for products:

$$\frac{\partial}{\partial \psi^7_H}\left(S^{\alpha\beta'}_{IJ}\psi^{\beta'}_J\right) = \frac{\partial S^{\alpha\beta'}_{IJ}}{\partial \psi^7_H}\psi^{\beta'}_J + S^{\alpha\beta'}_{IJ}\frac{\partial \psi^{\beta'}_J}{\partial \psi^7_H}. \tag{5.52}$$

Now,

$$\frac{\partial S_{IJ}^{\alpha\beta'}}{\partial \psi_H^\gamma} = \frac{\partial}{\partial \psi_H^\gamma}\left[(1-\theta)S_{IJ}^{\alpha\beta^n} + \theta S_{IJ}^{\alpha\beta(n+1)}\right]$$

$$= \frac{\partial}{\partial \psi_H^\gamma}\left[\theta S_{IJ}^{\alpha\beta(n+1)}\right]$$

$$= \theta \frac{\partial S_{IJ}^{\alpha\beta(n+1)}}{\partial \psi_H^\gamma} \tag{5.53}$$

We also know that, in general,

$$S_{IJ}^{\alpha\beta(n+1)} = (a_K)\int_{\Omega^e} N_K\left[N_J N_I\right]^T \, d\Omega \tag{5.54}$$

Thus

$$\frac{\partial S_{IJ}^{\alpha\beta(n+1)}}{\partial \psi_H^\gamma} = \frac{\partial (a_K)}{\partial \psi_H^\gamma}\int_{\Omega^e} N_K\left[N_J N_I\right]^T \, d\Omega$$

$$= \frac{\partial (a_K)}{\partial \psi_H^\gamma}\delta_{KH}\Upsilon_{KIJ}. \tag{5.55}$$

where Υ_{KIJ} represents the integral in Equation (5.54). This follows from the work in the previous subsection.

From Equation (5.52), and using the definition for $\psi_J^{\beta'}$ following from Equation (5.3),

$$S_{IJ}^{\alpha\beta'}\frac{\partial \psi_J^{\beta'}}{\partial \psi_H^\gamma} = S_{IJ}^{\alpha\beta'}\theta\delta_{JH}\delta^{\beta\gamma}$$

$$= \theta S_{IH}^{\alpha\gamma'}. \tag{5.56}$$

Again, the n-level term was constant and could be eliminated. The primary variables are orthogonal to each other so the value of the derivative is one for the same nodal point and type of degree of freedom, and zero for the rest. This condition is represented by the double-δ notation above. The term $S_{IH}^{\alpha\gamma'}$ is numerically equal to $S_{IJ}^{\alpha\beta'}$, but its use is governed by the different indices.

Thus we can write for the derivative of the first non-linear product

$$\frac{\partial}{\partial \psi_H^\gamma}\left(S_{IJ}^{\alpha\beta'}\psi_J^{\beta'}\right) = \theta\frac{\partial S_{IJ}^{\alpha\beta(n+1)}}{\partial \psi_H^\gamma}\psi_J^{\beta'} + \theta S_{IH}^{\alpha\gamma'} \tag{5.57a}$$

$$= \theta\frac{\partial S_{IJ}^{\alpha\beta(n+1)}}{\partial \psi_H^\gamma}\left[(1-\theta)\psi_J^{\beta^n} + \theta\psi_J^{\beta(n+1)}\right]$$

$$+ \left[\theta(1-\theta)S_{IH}^{\alpha\gamma^n} + \theta^2 S_{IH}^{\alpha\gamma(n+1)}\right]. \tag{5.57b}$$

Note that there is now only one derivative to be evaluated in the expression: that of $S_{IJ}^{\alpha\beta(n+1)}$. This is evaluated numerically.

The derivative of the second non-linear expression follows by a similar analysis to that of the first term:

$$\frac{\partial}{\partial\psi_H^\gamma}\left(D_{IJ}^{\alpha\beta'}\frac{\Delta\psi_J^{\beta^n}}{\Delta t_n}\right) = \frac{\partial D_{IJ}^{\alpha\beta'}}{\partial\psi_H^\gamma}\frac{\Delta\psi_J^{\beta^n}}{\Delta t_n} + \frac{D_{IJ}^{\alpha\beta'}}{\Delta t_n}\frac{\partial(\Delta\psi_J^{\beta^n})}{\partial\psi_H^\gamma}. \tag{5.58}$$

$D_{IJ}^{\alpha\beta'}$ is derived in the same manner as $S_{IJ}^{\alpha\beta'}$—as the sum over the element nodes of products of a variable coefficient at each node and a constant integrand, thus

$$\begin{aligned}
\frac{\partial D_{IJ}^{\alpha\beta'}}{\partial\psi_H^\gamma} &= \frac{\partial}{\partial\psi_H^\gamma}\left[(1-\theta)S_{IJ}^{\alpha\beta^n} + \theta D_{IJ}^{\alpha\beta(n+1)}\right] \\
&= \frac{\partial}{\partial\psi_H^\gamma}\left[\theta S_{IJ}^{\alpha\beta(n+1)}\right] \\
&= \theta\frac{\partial D_{IJ}^{\alpha\beta(n+1)}}{\partial\psi_H^\gamma} \\
&= \theta\frac{\partial(a_K)}{\partial\psi_H^\gamma}\int_{\Omega^e} N_K\,[N_J N_I]^T\,d\Omega \\
&= \theta\frac{\partial(a_K)}{\partial\psi_H^\gamma}\delta_{KH}\Upsilon_{KIJ}.
\end{aligned} \tag{5.59}$$

Now, because $\Delta\psi_J^{\beta^n} = \psi_J^{\beta(n+1)} - \psi_J^{\beta^n}$, the second term in Equation (5.58) is given by

$$\begin{aligned}
\frac{\partial(\Delta\psi_J^{\beta^n})}{\partial\psi_H^\gamma} &= \frac{\partial\psi_J^{\beta(n+1)}}{\partial\psi_H^\gamma} \\
&= \delta_{JH}\delta^{\beta\gamma}.
\end{aligned} \tag{5.60}$$

Therefore the derivative of the second non-linear product becomes

$$\begin{aligned}
\frac{\partial}{\partial\psi_H^\gamma}\left(D_{IJ}^{\alpha\beta'}\frac{\Delta\psi_J^{\beta^n}}{\Delta t_n}\right) &= \theta\frac{\partial D_{IJ}^{\alpha\beta(n+1)}}{\partial\psi_H^\gamma}\frac{\Delta\psi_J^{\beta^n}}{\Delta t_n} + \frac{D_{IJ}^{\alpha\beta'}}{\Delta t_n}\delta_{JH}\delta^{\beta\gamma} \\
&= \theta\frac{\partial D_{IJ}^{\alpha\beta(n+1)}}{\partial\psi_H^\gamma}\frac{\Delta\psi_J^{\beta^n}}{\Delta t_n} + \frac{D_{IH}^{\alpha\gamma'}}{\Delta t_n} && \text{(5.61a)} \\
&= \theta\frac{\partial D_{IJ}^{\alpha\beta(n+1)}}{\partial\psi_H^\gamma}\frac{\Delta\psi_J^{\beta^n}}{\Delta t_n} + \frac{1}{\Delta t_n}\left[(1-\theta)D_{IH}^{\alpha\gamma^n} + \theta D_{IH}^{\alpha\gamma(n+1)}\right]. && \text{(5.61b)}
\end{aligned}$$

Note that, as in the expansion of the derivative of the first non-linear term, there is now only one derivative to be evaluated in the expression: that of $D_{IJ}^{\alpha\beta(n+1)}$.

Analysis of the derivatives has yielded algebraic expressions that appear simpler to evaluate than the original forms. The amount of work to evaluate the derivative has been determined in each case, and these expansions require about one-half the number of operations needed by the original forms.

5.9 *Implementation of the discretized equations as a finite element computer program*

The finite elements used for this work are described completely in Appendix C. A full description of the implementation of the finite element program is given in Appendix D. Flow charts are provided to help describe the model. A practical well model with many of the options used in current finite difference models has been incorporated into the finite element scheme, described in Appendix D. The other major portions of the program are outlined: forming the (Newton-Raphson) Jacobian matrix, assembling the set of global equations, and renumbering the global mesh. Emphasis has been placed on the implementation of these algorithms for vector processing, commonly used to improve the time taken to perform the calculations. This has resulted in the development of a specialized Gaussian linear equation solver, and the use of pseudo-descriptors — a method of FORTRAN programming for vectors. Further description of the finite element program is found in the user manual contained in Appendix J.

5.10 *Discussion*

The finite element scheme developed here is a fully implicit approach which has not been presented before in the literature. The scheme describes a formal approach for assembling the equations of conservation of heat and mass in flow through a porous medium, and its subsequent coupling to a displacement model. The benefit of the totally finite element approach lies in the coincidence of the meshes for heat and fluid flow and the solid displacements, providing for complete compatibility of the solution domains.

It turns out that the approach is possibly not suitable for the solution of highly non-linear problems of this type. The model was only able to run two or three small timesteps before the pressure solutions began to diverge. This behaviour severely restricts the usefulness of the computer program. The development of the equations is exploited in developing the combined finite element and finite difference model in later chapters. The contribution of this work lies in the description of the finite element approach to this problem and presenting the difficulty of pursuing this route for a coupled solution. The novel material includes

(a) the description and preparation of the equations;

(b) the discretization in space of the equations;

(c) the discretization in time of the non-linear partial differential equations;

(d) the algebraic reductions of the derivatives used in the linearization scheme;

(e) the development of new correlations of physical properties of steam, water, and bitumen;

(f) the implementation of a practical well model for finite element reservoir simulation;

(g) the development of many new programming techniques to make the program run faster, e.g., the vectorized approach to the linear equation solver and the general use of arithmetic through pseudo-descriptors.

The formulation has been checked for errors and none was found, so the conclusion is that the method is not suitable for this class of problem. This is despite the use of upwinding techniques. However, the detailed formulation presented will be a groundwork for future discretization attempts using finite element methods. It is possible that the choice of time integration led to an unstable formulation.

Chapter VI

FINITE DIFFERENCE - FINITE ELEMENT APPROACH TO THE COUPLED MODEL

The large overhead in a fully finite element approach to the coupled problem of thermal fluid flow and solid deformation suggests that alternative approaches to the problem would be worthwhile to investigate. The natural progress of petroleum reservoir numerical simulation has been with finite difference rather than finite element discretization. However, finite differences have not been very successful for plastic deformation problems (e.g., Pyrah, 1987). A practical approach would be to combine solutions from a mixed discretization of finite differences and finite elements, even if it lacks the aesthetic aspects of a purely finite element scheme.

In this chapter two numerical models are described: a highly implicit finite difference thermal reservoir simulator, and a finite element elastoplastic solid model incorporating incremental plasticity theory. Various degrees of coupling are proposed to provide analyses of differing degrees of complexity commensurate with the computing resources required to run the models. Detailed information regarding the discretizations is given in Appendices E and F.

6.1 *A finite difference thermal fluid-solid coupled model*

The starting point for the model is the equations of continuity and accumulation for fluid flow and heat transport. The constitutive equations for flow — a molar balance — were already presented in Chapter IV, one for each component:

$$\vec{\nabla}\cdot\left(\phi\sum_{\alpha=o,w,g} x_{i\alpha}\hat{\rho}_\alpha \mathrm{v}_\alpha S_\alpha\right)+\frac{\partial}{\partial t}\left(\phi\sum_{\alpha=o,w,g} x_{i\alpha}\hat{\rho}_\alpha S_\alpha\right)=0. \qquad (6.1)$$

There are two components in a dead oil system, where no gas exists except for water vapour. In order to model a three phase system (oil-water-steam) a third equation for the vapour-liquid equilibrium of saturated water vapour is used:

$$T = f(p) \qquad (6.2)$$

Neglecting the contribution of the kinetic energy, potential energy, and viscous dissipation terms, the constitutive equation for heat transport can also be shown to be equal to (Abou-Kassem *et al.*, 1986)

$$\vec{\nabla}\cdot\left[\phi\sum_{\alpha=o,w,g} x_{i\alpha}\hat{\rho}_\alpha \mathbf{v}_\alpha S_\alpha H_\alpha + (1-\phi)\hat{\rho}_r \mathbf{v}_r H_r\right] + \vec{\nabla}\cdot(k_h T) + \frac{\partial}{\partial t}\left[\phi\sum_{\alpha=o,w,g} x_{i\alpha}\hat{\rho}_\alpha S_\alpha H_\alpha + (1-\phi)\hat{\rho}_r H_r\right] = 0. \quad (6.3)$$

Given the assumption of a dead oil system, the equations can be rewritten in terms of mass densities as there is only one component per phase. Furthermore, as the finite difference method requires mass conservation within a specified element of volume, source terms can be added to the equations. One ends up with the following four equations:

1. Energy equation:

$$\vec{\nabla}\cdot\left[\phi\sum_{\alpha=o,w,g} \rho_\alpha \mathbf{v}_\alpha S_\alpha H_\alpha + (1-\phi)\rho_r \mathbf{v}_r H_r\right] + \vec{\nabla}\cdot(k_h T) - q_L^* - q_H^* + \frac{\partial}{\partial t}\left[\phi\sum_{\alpha=o,w,g} \rho_\alpha S_\alpha H_\alpha + (1-\phi)\rho_r H_r\right] = 0. \quad (6.4)$$

where q_L^* is the rate of heat loss to the surrounding formations and q_H^* is the rate of production of enthalpy.

2. Water equation:

$$\vec{\nabla}\cdot[\phi(\rho_w \mathbf{v}_w S_w + \rho_g \mathbf{v}_g S_g)] - q_w^*\rho_w - q_g^*\rho_g + \frac{\partial}{\partial t}[\phi(\rho_w S_w + \rho_g S_g)] = 0. \quad (6.5)$$

3. Oil equation:

$$\vec{\nabla}\cdot(\phi\rho_o \mathbf{v}_o S_o) - q_o^*\rho_o + \frac{\partial}{\partial t}[\phi(\rho_o S_o] = 0. \quad (6.6)$$

where q_w^*, q_g^*, and q_o^* are respectively the rates of production of water, steam, and oil.

4. VLE equation

$$T = f(p_g). \quad (6.7)$$

The fluid velocities, $\mathbf{v}_\alpha$, in these equations comprise two parts: the fluid velocity, $\mathbf{V}_\alpha$, with respect to the rock or soil skeleton, and the rock or soil velocity, $\mathbf{v}_r$, with respect to a fixed reference point. As was shown in Chapter IV, this relation can be expressed by writing the microscopic velocity of a fluid phase α relative to the fixed frame of reference:

$$\mathbf{v}_\alpha = \frac{\mathbf{V}_\alpha}{\phi S_\alpha} + \mathbf{v}_r \quad (6.8)$$

or

$$\mathbf{V}_\alpha = \phi S_\alpha(\mathbf{v}_\alpha - \mathbf{v}_r) \tag{6.9}$$

where $\mathbf{V}_\alpha$ is the Darcy (bulk macroscopic) fluid velocity with respect to the soil skeleton, and $\mathbf{v}_r$ is the microscopic soil velocity. The Darcy velocity is given by

$$\mathbf{V}_\alpha = -\lambda_\alpha \mathbf{k} \cdot (\nabla p_\alpha + \rho_\alpha \mathbf{g}) \tag{6.10}$$

The heat and flow equations can now be rewritten as:

1. Energy equation:

$$\vec{\nabla}\cdot\left[\sum_{\alpha=o,w,g} \rho_\alpha(\mathbf{V}_\alpha + \phi S_\alpha \mathbf{v}_r)H_\alpha + (1-\phi)\rho_r \mathbf{v}_r H_r\right] + \vec{\nabla}\cdot(k_h T) - q_L^{\bullet} - q_H^{\bullet} + \frac{\partial}{\partial t}\left[\phi \sum_{\alpha=o,w,g} \rho_\alpha S_\alpha H_\alpha + (1-\phi)\rho_r H_r\right] = 0. \tag{6.11}$$

2. Water equation:

$$\vec{\nabla}\cdot[\rho_w(\mathbf{V}_w + \phi S_w \mathbf{v}_r) + \rho_g(\mathbf{V}_g + \phi S_g \mathbf{v}_r)] - q_w^{\bullet}\rho_w - q_g^{\bullet}\rho_g + \frac{\partial}{\partial t}[\phi(\rho_w S_w + \rho_g S_g] = 0. \tag{6.12}$$

3. Oil equation:

$$\vec{\nabla}\cdot[\rho_o(\mathbf{V}_o + \phi S_o \mathbf{v}_r)] - q_o^{\bullet}\rho_o + \frac{\partial}{\partial t}(\phi \rho_o S_o) = 0. \tag{6.13}$$

All three equations can now be expanded so that the terms containing soil velocities can be separated out. The time rate of change of porosity is actually related to the solid velocity (see later sections), and is considered to be a "solid" term. The following sections extract the solid terms from each equation and expand them so that they can be incorporated in the finite difference discretization.

6.2 *Solid terms in the heat equation*

The solid terms are considered separately for the convection and accumulation portions of the equation.

6.2.1 *Convection terms*

The general convection term of the heat equation is given by:

$$\vec{\nabla}\cdot\left[\sum_{\alpha=o,w,g} \rho_\alpha(\mathbf{V}_\alpha + \phi S_\alpha \mathbf{v}_r)H_\alpha + (1-\phi)\rho_r \mathbf{v}_r H_r\right] \tag{6.14}$$

Two terms contributed by the solid phase can be identified as:

$$\vec{\nabla}\cdot\left(\sum_{\alpha=o,w,g}\rho_\alpha\phi S_\alpha \mathbf{v}_r H_\alpha\right)+\vec{\nabla}\cdot\left[(1-\phi)\rho_r\mathbf{v}_r H_r\right] \tag{6.15}$$

These terms can be each further expanded to give three new terms, by selective differentiation. For the first term one can write

$$\phi\mathbf{v}_r\cdot\vec{\nabla}\left(\sum_{\alpha=o,w,g}\rho_\alpha S_\alpha H_\alpha\right) \tag{6.16a}$$

$$\phi\left(\sum_{\alpha=o,w,g}\rho_\alpha S_\alpha H_\alpha\right)\vec{\nabla}\cdot\mathbf{v}_r \tag{6.16b}$$

$$\left(\sum_{\alpha=o,w,g}\rho_\alpha S_\alpha H_\alpha\right)\mathbf{v}_r\cdot\vec{\nabla}\phi \tag{6.16c}$$

and the second term:

$$(1-\phi)\mathbf{v}_r\cdot\vec{\nabla}(\rho_r H_r) \tag{6.17a}$$

$$(1-\phi)\rho_r H_r\vec{\nabla}\cdot\mathbf{v}_r \tag{6.17b}$$

$$-\rho_r H_r\mathbf{v}_r\cdot\vec{\nabla}\phi \tag{6.17c}$$

It should also be noted that the term $\rho_r H_r$ can be expanded in terms of the specific heat capacity of the rock, assuming c_{pr} is constant in the range of ΔT,

$$\rho_r H_r=\rho_r c_{pr}\Delta T \tag{6.18}$$

6.2.2 *Accumulation terms*

The general accumulation term in the heat equation is given by

$$\frac{\partial}{\partial t}\left[\phi\sum_{\alpha=o,w,g}\rho_\alpha S_\alpha H_\alpha+(1-\phi)\rho_r H_r\right] \tag{6.19}$$

The two main terms contributed by the solid matrix to this expression are:

$$\frac{\partial}{\partial t}\left[(1-\phi)\rho_r H_r\right]+\sum_{\alpha=o,w,g}\rho_\alpha S_\alpha H_\alpha\frac{\partial\phi}{\partial t} \tag{6.20}$$

Traditionally, $\frac{\partial\phi}{\partial t}$ is taken care of through a lumped parameter known as pore compressibility. This is essentially the calculated elastic response of the matrix to a given change in confining stress, resulting from a change in the pore pressure. It will now be handled directly from the calculated volumetric strains.

The first term is now expanded into two further terms:

$$(1-\phi)\frac{\partial}{\partial t}(\rho_r H_r) - \rho_r H_r \frac{\partial \phi}{\partial t} \tag{6.21}$$

The term including the time derivative of $\rho_r H_r$ can be expanded considering Equation (6.18), assuming that c_{pr} is a constant and that the solid grains are incompressible. The latter assumption implies that ρ_r is constant with changing pressure, a common assumption in soil and rock mechanics (Huyakorn and Pinder, 1987). The individual solid grains change volume with a change in temperature, which can be accounted for using the isobaric bulk coefficient of thermal expansion (assumed constant), where

$$\begin{aligned} \alpha &= \frac{1}{v}\frac{\partial v}{\partial T} \\ &= -\frac{1}{\rho}\frac{\partial \rho}{\partial T} \end{aligned} \tag{6.22}$$

A first order expansion of the derivative with respect to T yields the following expression relating ρ to some reference value $\rho_0(T_0)$.

$$\rho = \rho_0(1-\alpha\Delta T) \tag{6.23}$$

This approximation is now differentiated with respect to T to find:

$$\frac{\partial \rho}{\partial T} = -\rho_0\alpha \tag{6.24}$$

Thus one can write:

$$\begin{aligned} (1-\phi)\frac{\partial}{\partial t}(\rho_r H_r) &= (1-\phi)\left[\rho_r\frac{\partial H_r}{\partial t} + H_r\frac{\partial \rho_r}{\partial t}\right] \\ &= (1-\phi)\left[\rho_r c_{pr}\frac{\partial T}{\partial t} + H_r\frac{\partial \rho_r}{\partial T}\frac{\partial T}{\partial t}\right] \\ &= (1-\phi)(\rho_r c_{pr} - H_r\rho_r\alpha)\frac{\partial T}{\partial t} \end{aligned} \tag{6.25}$$

The term containing the time derivative of porosity, $\frac{\partial \phi}{\partial t}$, can be expanded using the expression for porosity terms found in §6.4.1. A direct method is found as

$$\frac{\partial \phi}{\partial t} \approx \frac{1-\phi}{1+\epsilon_v}\frac{(\epsilon_v - \alpha\Delta T)}{\Delta t} \tag{6.26}$$

A general expression for porosity as a function of some reference value $\phi_0(p_0, T_0)$ is also presented from §6.4.1:

$$\phi^{n+1} \approx \frac{\phi_0 + \varepsilon_v^{n+1} - (1-\phi_0)\alpha(T^{n+1} - T_0)}{1 + \varepsilon_v^{n+1}} \tag{6.27}$$

where ε_v^{n+1} is the incremental volumetric strain since the departure from the initial condition of ϕ_0. Whence one can solve the difference expression

$$\frac{\partial \phi}{\partial t} \approx \frac{\phi^{n+1} - \phi^n}{\Delta t} \tag{6.28}$$

6.3 *Solid terms in the fluid flow equations*

The solid terms are considered separately for the convection and accumulation portions of the equation.

6.3.1 *Convection terms*

The general convection term in any of the fluid equations is given by:

$$\vec{\nabla} \cdot [\rho_\alpha(\mathbf{V}_\alpha + \phi S_\alpha \mathbf{v}_r)]\,. \tag{6.29}$$

where α is the phase in question. There are two such terms in the water equation (one each for water and gas) and one in the oil equation. The term contributed by the solid phase is identifiable as:

$$\vec{\nabla} \cdot (\phi \rho_\alpha S_\alpha \mathbf{v}_r)\,. \tag{6.30}$$

This space derivative can be expanded to

$$\vec{\nabla} \cdot (\phi \rho_\alpha S_\alpha \mathbf{v}_r) = \phi \mathbf{v}_r \cdot \vec{\nabla}(\rho_\alpha S_\alpha) + \phi \rho_\alpha S_\alpha \vec{\nabla} \cdot \mathbf{v}_r + \rho_\alpha S_\alpha \mathbf{v}_r \cdot \vec{\nabla}\phi. \tag{6.31}$$

Each of these terms can now be found as follows. The first term is

$$\phi \mathbf{v}_r \cdot \vec{\nabla}(\rho_\alpha S_\alpha) = \phi\,[v_x \Delta_x(\rho_\alpha S_\alpha) + v_y \Delta_y(\rho_\alpha S_\alpha) + v_z \Delta_z(\rho_\alpha S_\alpha)] \tag{6.32}$$

where $\Delta_x, \Delta_y, \Delta_z$ are the appropriate difference operators. The second term is

$$\phi \rho_\alpha S_\alpha \vec{\nabla} \cdot \mathbf{v}_r = \phi \rho_\alpha S_\alpha \frac{\Delta \varepsilon_v}{\Delta t} \tag{6.33}$$

where the expansion of $\vec{\nabla} \cdot \mathbf{v}_r$ is given in a later section. The third term is given by

$$\rho_\alpha S_\alpha \mathbf{v}_r \cdot \vec{\nabla}\phi = \rho_\alpha S_\alpha(v_x \Delta_x \phi + v_y \Delta_y \phi + v_z \Delta_z \phi) \tag{6.34}$$

where the expansion is as before with difference operators.

6.3.2 *Accumulation terms*

The general accumulation term for phase α in the flow equations is

$$\frac{\partial}{\partial t}(\phi \rho_\alpha S_\alpha) \tag{6.35}$$

There are two such terms in the water equation (one each for water and gas) and one in the oil equation. The term is simply expanded as

$$\frac{\partial}{\partial t}(\phi \rho_\alpha S_\alpha) = \phi \frac{\partial}{\partial t}(\rho_\alpha S_\alpha) + \rho_\alpha S_\alpha \frac{\partial \phi}{\partial t} \tag{6.36}$$

This second term is effectively a solid term, as the change in porosity is a function of the volumetric strain. Traditionally in reservoir simulation it is handled through a pore compressibility term which relates change in pressure to change in porosity. The pore compressibility is sometimes confused with the bulk rock compressibility, a point discussed by Geertsma (1957) and reviewed by Settari (1989). A rigorous definition of change in porosity as a function of volumetric strain can be derived as shown later in §6.4.1, from which an equivalent pore compressibility term can be extracted.

6.4 *Evaluation of the solid terms*

The solid coupling terms were presented in the preceding section. The question remains how the expressions to evaluate them were obtained. These are presented here.

6.4.1 *Porosity terms*

The definition of porosity is

$$\phi = \frac{V_b - V_r}{V_b} \tag{6.37}$$

where V_b is the bulk volume of the rock and V_r is the rock (matrix) volume. After some change in pore pressure and temperature to (p, T) from the initial conditions of $\phi_0(p_0, T_0)$, incurring an increment of volumetric strain ε_v, one obtains a new porosity,

$$\begin{aligned}\phi &= \frac{(V_b + V_b\varepsilon_v) - (V_r + \Delta V_r)}{V_b + V_b\varepsilon_v} \\ &= \frac{1}{1+\varepsilon_v}\left[\frac{(1+\varepsilon_v)V_b - (V_r + \Delta V_r)}{V_b}\right] \\ &= \frac{1}{1+\varepsilon_v}\left[\frac{V_b - V_r}{V_b} + \varepsilon_v - \frac{\Delta V_r}{V_b}\right] \\ &= \frac{1}{1+\varepsilon_v}\left[\phi_0 + \varepsilon_v - \frac{\Delta V_r}{V_b}\right] \qquad (6.38)\end{aligned}$$

where $\Delta V_b = V_b\varepsilon_v$ was implied. The change in rock volume can be attributed entirely to thermal expansion if the rock grains are considered incompressible.

$$\Delta V_r = V_r\alpha(T - T_0) \qquad (6.39)$$

Furthermore,

$$\begin{aligned}\frac{V_b - V_r}{V_b} - 1 &= -\frac{V_r}{V_b} \\ &= \phi_0 - 1 \qquad (6.40)\end{aligned}$$

Using these last two relationships in Equation (6.38), one obtains

$$\phi = \frac{1}{1+\varepsilon_v}\left[\phi_0 + \varepsilon_v - (1-\phi_0)\alpha(T - T_0)\right] \qquad (6.41)$$

For small changes in porosity, the derivative with respect to time can be found using the forward difference

$$\begin{aligned}\frac{\partial\phi}{\partial t} &\approx \frac{\phi - \phi_0}{\Delta t} \\ &= \frac{1}{1+\varepsilon_v}\frac{\phi_0 + \varepsilon_v - (1-\phi_0)\alpha\Delta T - \phi_0(1+\varepsilon_v)}{\Delta t} \\ &= \frac{1-\phi_0}{1+\varepsilon_v}\frac{\varepsilon_v - \alpha\Delta T}{\Delta t} \qquad (6.42)\end{aligned}$$

where $\Delta T = T - T_0$.

6.4.2 *Relating pore compressibility to material changes*

Pore compressibility is rigorously defined as:

$$c_\phi = \frac{1}{\phi}\frac{\partial\phi}{\partial p} \qquad (6.43)$$

Note that this is not the same as the bulk rock compressibility, or the individual grain compressibilities which are considered negligible in this work. The appropriate integration, assuming c_ϕ to be constant over the interval, leads to the following expression for porosity:

$$\phi = \phi_0 \exp^{c_\phi (p - p_0)} \tag{6.44}$$

A first order approximation to this expression is given by

$$\phi = \phi_0 [1 + c_\phi \Delta p] \tag{6.45}$$

where $\Delta p = p - p_0$. This expression can now be compared to the earlier expression for porosity, Equation (6.41), and solved for c_ϕ:

$$c_\phi = \frac{(1 - \phi_0)(\Delta\epsilon_v - \alpha\Delta T)}{(1 + \Delta\epsilon_v)\phi_0 \Delta p} \tag{6.46}$$

Beattie *et al.* (1991) pointed out that for relatively large deformations the first order approximation for the pore compressibility as a function of pressure would be inaccurate. Accordingly, an expression for c_ϕ obtained from the full solution is given by

$$c_\phi = \frac{\ln\left[\dfrac{\phi_0 + \Delta\epsilon_v - (1 - \phi_0)\alpha\Delta T}{\phi_0(1 + \Delta\epsilon_v)}\right]}{\Delta p} \tag{6.47}$$

6.4.3 *Relating dilation to changes in permeability*

Amyx, Bass and Whiting (1960) presented an expression for permeability as the Kozeny equation, from the analysis of a bundle of capillaries:

$$k = \frac{\phi}{k_Z S_p^2} \tag{6.48}$$

where $k_Z \approx 5$ is the Kozeny constant (actually a function of the shape factor and tortuosity of the porous medium), and S_p is the specific surface area (per unit volume of pore space):

$$S_p = \frac{A_s}{V_p} \tag{6.49}$$

The Kozeny relation is not necessarily the best approach to finding an expression for permeability, but it provides a means to examine the relation between the deformation of a rock skeleton and changes in the permeability. It will be incorporated in the model as the relation between changes in volumetric strain and permeability, but the model will not be restricted to this particular relationship. A real study would demand laboratory determination of the changes in permeability associated with deformation of the sand.

Consider a change in conditions from k_0, S_{p_0}, A_{s_0} at (p_0, T_0) to some arbitrary state (p, T). There will be a change in the bulk volume and the cumulative volume of the individual grains. There will also be change in the surface area of the grains. The changes in grain volume and area are assumed to result only from thermal expansion or contraction, so that the grains are considered to be incompressible. The change in surface area can be found by considering the grains to be spherical and calculating the change in the radius as a result of thermal expansion or contraction:

$$\Delta A_s = n\frac{8}{3}\pi r^2 \alpha \Delta T \tag{6.50}$$

where the grains are assumed to have a uniform and constant initial radius r and there are n grains per unit volume of pore space. The increment of specific surface area ΔA_s will be denoted by β:

$$A_s = A_{s_0}(1+\beta) \tag{6.51}$$

The β term will be carried through the analysis, but is not actually considered in the final implementation, as it is assumed that β can be neglected because of the lesser order of change compared to the change in volume. The change in volume can be represented by

$$\Delta V_r = V_r \alpha \Delta T \tag{6.52}$$

where V_r is the rock grain volume. Therefore any change in rock bulk volume, less the thermal expansion term, will represent a change in the void volume:

$$\Delta V_p = \Delta V - \Delta V_r \tag{6.53}$$

Thus the new porosity will be given by

$$\phi = \frac{V_p + (\Delta V - \Delta V_r)}{V_b + \Delta V} \tag{6.54}$$

The new value of S_p is given by

$$S_p = \frac{A_{s_0}(1+\beta)}{V_p + (\Delta V - \Delta V_r)} \tag{6.55}$$

Thus the ratio of the new to original permeability is given by

$$\begin{aligned} \frac{k}{k_0} &= \frac{\phi}{\phi_0}\frac{S_{p_0}^2}{S_p^2} \\ &= \left[\frac{V_p + \Delta V_p}{V_b + \Delta V_b}\frac{(V_p + \Delta V_p)^2}{A_{s_0}^2(1+\beta)^2}\right]\left[\frac{V_b}{V_p}\frac{A_{s_0}^2}{V_p^2}\right] \end{aligned} \tag{6.56}$$

Now, the change in bulk volume is found directly from the volumetric strain:

$$\Delta V_b = \epsilon_v V_b \tag{6.57}$$

Hence one can write

$$\begin{aligned}\frac{k}{k_0} &= \frac{(V_p + \Delta V_p)^3}{V_b(1+\epsilon_v)(1+\beta)^2}\frac{V_b}{V_p^3} \\ &= \frac{1}{(1+\epsilon_v)}\frac{1}{(1+\beta)^2}\left(\frac{V_p+\Delta V_p}{V_p}\right)^3\end{aligned} \tag{6.58}$$

Now, expanding the change in pore volume:

$$\begin{aligned}\Delta V_p &= \Delta V - \Delta V_r \\ &= \epsilon_v V_b - V_r \alpha \Delta T \\ &= \epsilon_v V_b - (V_b - V_p)\alpha\Delta T\end{aligned} \tag{6.59}$$

Therefore,

$$V_p + \Delta V_p = V_p(1+\alpha\Delta T) + V_b(\epsilon_v - \alpha\Delta T) \tag{6.60}$$

and,

$$\begin{aligned}\frac{V_p+\Delta V_p}{V_p} &= (1+\alpha\Delta T) + \frac{\epsilon_v - \alpha\Delta T}{\phi_0} \\ &= \frac{\epsilon_v + \alpha\Delta T(\phi_0 - 1) + \phi_0}{\phi_0} \\ &= 1 + \frac{\epsilon_v}{\phi_0} - \frac{\alpha\Delta T(1-\phi_0)}{\phi_0}\end{aligned} \tag{6.61}$$

Use this result in the expression for the change in permeability;

$$\frac{k}{k_0} = \frac{\left[1 + \frac{\epsilon_v}{\phi_0} - \frac{\alpha\Delta T(1-\phi_0)}{\phi_0}\right]^3}{(1+\epsilon_v)(1+\beta)^2} \tag{6.62}$$

which includes the thermal effects. If one assumes that change in grain surface area is small compared to the volumetric changes, the expression simplifies to

$$\frac{k}{k_0} = \frac{\left[1 + \frac{\epsilon_v}{\phi_0} - \frac{\alpha\Delta T(1-\phi_0)}{\phi_0}\right]^3}{1+\epsilon_v} \tag{6.63}$$

Ignoring the thermal effects entirely leads to the following relatively trivial expression:

$$\frac{k}{k_0} = \frac{\left(1 + \frac{\varepsilon_v}{\phi_0}\right)^3}{1 + \varepsilon_v} \tag{6.64}$$

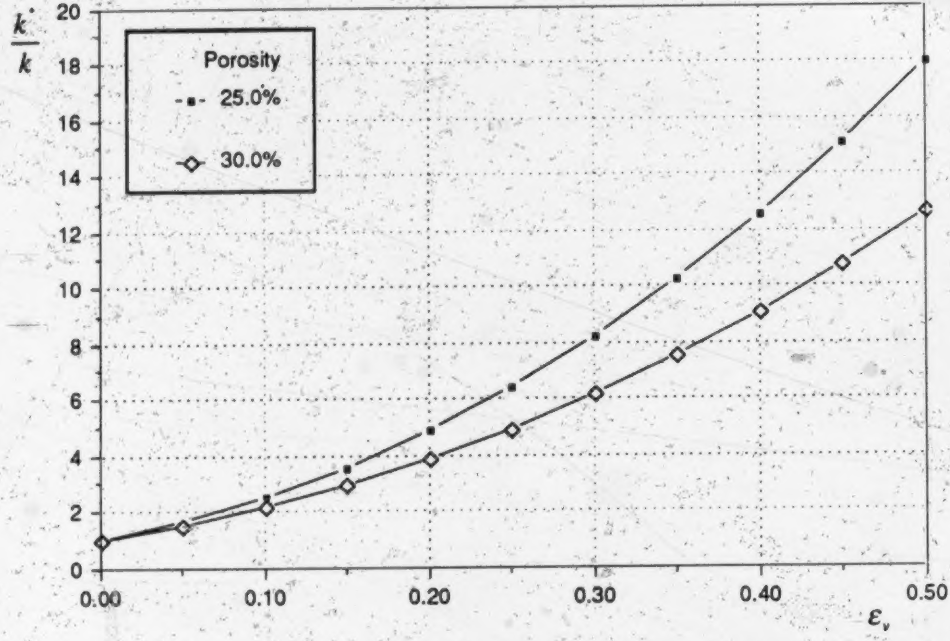


Figure 6–1

Permeability ratio as a function of volumetric strain according to Kozeny-Poiseuille model.

This expression is a first attempt at systematically incorporating volumetric strains into the (absolute) permeability. The relation between changes in permeability as a function of volumetric strain and porosity, calculated from Equation (6.64), is shown graphically in Figure 6–1. In a sand body with a normal distribution of sand grain sizes, however, the permeability could go down with increasing deformation. This would be due to the mobilization of the smaller particles and their subsequent bridging and plugging of pore throats.

6.4.4 *Evaluation of the space derivative terms*

Various space derivatives must be evaluated to complete the solid terms. The term $\vec{\nabla} \cdot \mathbf{v}_r$ is found directly as follows:

$$\begin{aligned}
\vec{\nabla} \cdot \mathbf{v}_r &= \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z} \right) \cdot (v_x, v_y, v_z) \\
&= \frac{\partial}{\partial x} \frac{\partial u_x}{\partial t} + \frac{\partial}{\partial y} \frac{\partial u_y}{\partial t} + \frac{\partial}{\partial z} \frac{\partial u_z}{\partial t} \\
&= \frac{\partial}{\partial t} \left(\frac{\partial u_x}{\partial x} + \frac{\partial u_y}{\partial y} + \frac{\partial u_z}{\partial z} \right) \\
&= \frac{\partial}{\partial t} (\epsilon_{xx} + \epsilon_{yy} + \epsilon_{zz}) \\
&= \frac{\partial}{\partial t} [\mathrm{Tr}(\epsilon_{ij})] \\
&= \frac{\partial \epsilon_v}{\partial t} \\
&\approx \frac{\Delta \epsilon_v}{\Delta t} \qquad (6.65)
\end{aligned}$$

Other terms must be evaluated using finite differences. Where possible, central differences are used, where the first derivative $u'(x)$ with respect to x is found as

$$u'(x) = \frac{u_{i+1} - u_{i-1}}{\Delta x_{i-\frac{1}{2}} + \Delta x_{i+\frac{1}{2}}} \qquad (6.66)$$

where the $i - \frac{1}{2}$ and $i + \frac{1}{2}$ locations are the block boundaries in the $-x$ and $+x$ directions respectively, and

$$\Delta x_{i\pm\frac{1}{2}} = \frac{\Delta x_i + \Delta x_{i\pm1}}{2} \qquad (6.67)$$

In other words, $\Delta x_{i\pm\frac{1}{2}}$ is the distance between the current and neighbouring block centres. A similar approach is taken with the y and z directions. On boundaries, the appropriate forward or backward difference formulation must be used:

$$u'(x) \approx \frac{u_{i+1} - u_i}{\Delta x_{i+\frac{1}{2}}} \qquad (6.68a)$$

$$u'(x) \approx \frac{u_i - u_{i-1}}{\Delta x_{i-\frac{1}{2}}} \qquad (6.68b)$$

6.5 *A finite element model for evaluating elastoplastic deformations*

The development of the elastoplastic finite element model PLAST-PT3 is described more fully in Appendix F. The main outline of the model is provided here, as well as some detailed descriptions of novel features.

6.5.1 *Formulation and statement of the equations of equilibrium*

By convention in continuum mechanics, stresses are positive when in tension and negative when in compression. The convention in applied geomechanics is to reverse this notation, as the majority of stresses in this field are in compression. This work will retain the tension-positive convention of continuum mechanics due to the extensive algebra used. Furthermore, tensor (Einstein) notation is used extensively.

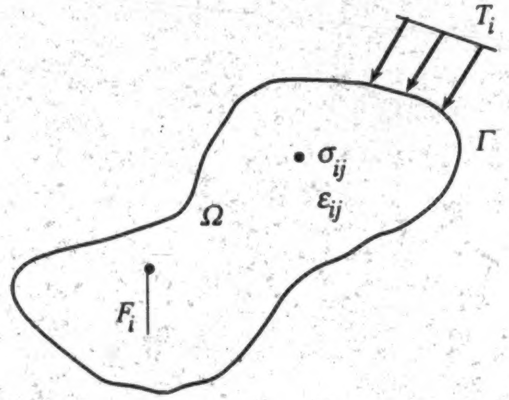


Figure 6–2

Forces acting on an arbitrary body.

Effective stress is given by

$$\sigma'_{ij} = \sigma_{ij} + \delta_{ij} p \tag{6.69}$$

where the Biot parameter α' has been dropped as it is considered to be equal to one in systems with incompressible particles. (See assumption in earlier sections.) Consider now an arbitrary body Ω, shown in Figure 6–2, subject to boundary tractions T_i on some portion of the surface Γ, internal body forces F_i, and a resultant stress field σ_{ij} and strains ϵ_{ij}. In any system, energy must be conserved, and in the absence of any other energy equivalents, there must be an equivalence of internal and external work, which can be stated as

$$\int_\Omega \sigma_{ij} \delta\epsilon_{ij} \, d\Omega = \int_\Omega F_i \delta u_i \, d\Omega + \int_\Gamma T_i \delta u_i \, d\Gamma \tag{6.70}$$

where δu_i are the displacements arising from the applied loads. This statement can be rewritten in terms of effective stress

$$\sigma_{ij} = \sigma'_{ij} - \delta_{ij} p \tag{6.71}$$

Whence:

$$\int_\Omega \sigma'_{ij} \delta\varepsilon_{ij} \, d\Omega = \int_\Omega F_i \delta u_i \, d\Omega + \int_\Gamma T_i \delta u_i \, d\Gamma + \int_\Omega F_i \delta_{ij} p \delta\varepsilon_{ij} \, d\Omega \tag{6.72}$$

These equations can be recast in vector form, using the symmetric nature of the stress and strain tensors. Given the full form of the stress tensor

$$\sigma_{ij} \equiv \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{bmatrix} \tag{6.73}$$

and that $\sigma_{xy} = \sigma_{yx}$, $\sigma_{yz} = \sigma_{zy}$, and $\sigma_{zx} = \sigma_{xz}$, then the vector form is given by

$$\underline{\sigma} = \langle \sigma_{xx}, \sigma_{yy}, \sigma_{zz}, \sigma_{yz}, \sigma_{zx}, \sigma_{xy} \rangle^T \tag{6.74}$$

The vector form of the strain tensor is found similarly. Note the relationship between stress and strain:

$$\underline{\sigma}' = \mathbf{C}\underline{\varepsilon} \tag{6.75}$$

where $\mathbf{C}$ is the constitutive matrix, and the strains are related to the nodal displacements $\bar{u}$:

$$\underline{\varepsilon} \approx \mathbf{L}\bar{u} \tag{6.76}$$

where $\mathbf{L}$ is the matrix of space derivative operators. The Kronecker delta is written as:

$$\delta_{ij} = \underline{m} = \langle 1, 1, 1, 0, 0, 0 \rangle^T \tag{6.77}$$

The equilibrium equation can now be written in vector notation as

$$\int_\Omega \underline{\sigma}' : \delta\underline{\varepsilon} \, d\Omega = \int_\Omega \underline{F} \cdot \delta\underline{u} \, d\Omega + \int_\Gamma \underline{T} \cdot \delta\underline{u} \, d\Gamma + \int_\Omega p\underline{m}\delta\underline{\varepsilon} \, d\Omega \tag{6.78}$$

Now make the following substitutions for the incremental displacements and strains. For a finite element, the displacements can be approximated by the product of the shape functions and the incremental displacements at the nodes:

$$\delta\underline{u} \approx \underline{N}\delta\bar{u} \tag{6.79}$$

and

$$\delta\underline{\varepsilon} \approx \mathbf{B}\delta\bar{u} \tag{6.80}$$

where $\mathbf{B}$ is the matrix of space derivatives of shape functions, $\mathbf{B} = \mathbf{L}\underline{N}$. On insertion of these terms, Equation (6.78) becomes

$$\int_\Omega \mathbf{C} : \mathbf{B} : \bar{\mathbf{u}} \mathbf{B} \delta \bar{\mathbf{u}} \, d\Omega = \int_\Omega \underline{F} \cdot \underline{N} \delta \bar{\mathbf{u}} \, d\Omega + \int_\Gamma \underline{T} \cdot \underline{N} \delta \bar{\mathbf{u}} \, d\Gamma + \int_\Omega p \underline{m} \mathbf{B} \delta \bar{\mathbf{u}} \, d\Omega \tag{6.81}$$

The values of $\delta \boldsymbol{u}$ are arbitrary and may be cancelled through Equation (6.81), giving

$$\left[\int_\Omega \mathbf{B}^T \mathbf{C} \mathbf{B} \, d\Omega \right] \bar{u} = \int_\Omega \underline{N}^T \underline{F} \, d\Omega + \int_\Gamma \underline{N}^T \underline{T} \, d\Gamma + \int_\Omega \underline{m} \mathbf{B} p \, d\Omega \tag{6.82}$$

which is an effective stress formulation of the equilibrium equations. This can also be expressed as the equilibrium of named loads:

$$[\mathbf{K}] \bar{\boldsymbol{u}} = \{F_{body}\} + \{F_{traction}\} + \{F_{pore}\} \tag{6.83}$$

Thermal expansion is incorporated as an initial strain, generating the stress counterpart, an initial stress:

$$\underline{\sigma}_0 = \mathbf{C} \underline{\varepsilon}_0 \tag{6.84}$$

Thus the total effective stress is given by

$$\underline{\sigma}' = \underline{\sigma}_0 + \Delta \underline{\sigma}' \tag{6.85}$$

and one can define a new load term for the initial strain:

$$F_{\sigma_0} \delta \bar{\mathbf{u}} = \int_\Omega \underline{\sigma}_0 \mathbf{B} \delta \bar{\mathbf{u}} \, d\Omega \tag{6.86}$$

or

$$F_{\sigma_0} = \int_\Omega \mathbf{B}^T \underline{\sigma}_0 \, d\Omega \tag{6.87}$$

which is part of the sum on the left-hand side of Equation (6.82). Thus, the equilibrium equation becomes

$$[\mathbf{K}] \bar{\mathbf{u}} = \{F_{body}\} + \{F_{traction}\} + \{F_{pore}\} - \{F_{\sigma_0}\} \tag{6.88}$$

The incorporation of plasticity into the stiffness matrix [K] is discussed in detail in Appendix F.

6.5.2 *Smoothing data to mesh nodes*

Most parameter values in a finite element model are found at the Gauss quadrature points so that the integrals can be evaluated. The only variables found at the nodes are the primary solution variables, the displacements (u, v, w). However, other parameters are required to quantify the behaviour of the deformed structure, such as the stress tensor, volumetric strain, and effective plastic strain, and to relate them to the fluid model. In particular, the principal stresses and their directions are of interest as they indicate when and where a hydraulic parting might occur, and its orientation. Two methods were investigated for smoothing values at the Gauss (quadrature) points to the mesh nodes: a global and a local scheme.

Both smoothing approaches treat the problem as one of finding an approximation to a known function, ϕ. Let the approximation be $\hat{\phi}$, then the residual over the domain Ω can be written as

$$R_\Omega = \phi - \hat{\phi} \tag{6.89}$$

The weighted residual statement (see Appendix C) can then be written, such that

$$\begin{aligned}\int_\Omega W_\ell(\phi - \hat{\phi})\, d\Omega &\equiv \int_\Omega W_\ell R_\Omega \, d\Omega \\ &= 0\end{aligned} \tag{6.90}$$

Let the approximation be made from a combination of trial functions N_m and constant parameters a_m, the latter chosen to optimize the approximation, i.e., minimize the residual:

$$\phi \approx \hat{\phi} = a_m N_m, \quad m = 1, \ldots, M \tag{6.91}$$

Note that index notation is used, and repeated indices indicate summation is to take place. Inserting this approximation:

$$\begin{aligned}\int_\Omega W_\ell(\phi - \hat{\phi})\, d\Omega &= \int_\Omega W_\ell(\phi - a_m N_m)\, d\Omega \\ &= 0\end{aligned} \tag{6.92}$$

On rearrangement, this becomes:

$$\int_\Omega W_\ell a_m N_m \, d\Omega = \int_\Omega W_\ell \phi \, d\Omega \tag{6.93}$$

or, simply,

$$\underline{\underline{K}}\, \underline{a} = \underline{f} \tag{6.94}$$

where

$$\left.\begin{aligned} k_{\ell m} &= \int_\Omega W_\ell N_m \, d\Omega \\ \langle a \rangle &= (a_1, a_2, \ldots, a_m) \\ f_\ell &= \int_\Omega W_\ell \phi \, d\Omega \end{aligned}\right\} \tag{6.95}$$

Let ϕ represent the smooth curve, surface, etc., of values of $\underline{\sigma}$, ϵ_v, or $\bar{\epsilon}_p$. These values are only actually known at discrete positions — coinciding with the quadrature points used for the solid model integrations. If the same quadrature scheme is chosen for solving this new approximation problem, the problem can be solved because the function ϕ is known at these points. Whence the right-hand side of the equation can be found, and thus the values of a_m, which can be nodal values.

This scheme has been shown for a single domain. For the global smoothing approach, the usual discretization into subdomains and approximations by piecewise trial functions, and element assembly, can be made:

$$\sum_{e=1}^{E}\int_{\Omega^e} W_\ell a_m N_m \, d\Omega = \sum_{e=1}^{E}\int_{\Omega^e} W_\ell \phi \, d\Omega \tag{6.96}$$

The choice of weighting function W_ℓ is important because only a limited number of data points are known on the surface being fitted. The known points would ideally coincide with the positions of the required quadrature points. The main choices — point collocation, subdomain collocation, Galerkin method, and least squares method — are described in Zienkiewicz and Morgan (1983). The Galerkin method was chosen because of the use of one function for both interpolation and weighting and the resultant symmetry in the coefficient matrix. The assignment $W_\ell = N_\ell$ is made; thus

$$\left.\begin{aligned} k_{\ell m} &= \int_\Omega N_\ell N_m \, d\Omega \\ f_\ell &= \int_\Omega N_\ell \phi \, d\Omega \end{aligned}\right\} \tag{6.97}$$

This method has coefficients requiring straightforward integration over the domain Ω. The values of a_m found are the nodal values of $\underline{\sigma}$, ϵ_v, or $\bar{\epsilon}_p$ satisfying the weighted residual statement. The least squares approach returns the same values, as is the result reported by Hinton and Campbell (1974), who bypassed the weighted residual statement.

Tests on the global smoothing method showed that it gives quite unrealistic values at the boundary nodes, requiring some correction. Instead, a series of local approximations can be made, and the average value at each node can be taken. The approach, using the weighted residual statement and the Galerkin method, remains the same. However, the approximation is now confined to each element, which is solved for in each case, rather than an assembly of elements. Now one must solve every element equation separately:

$$\underline{\underline{K}}\,\underline{a} = \underline{f}$$

The solution of this system of equations is trivial for a single element, compared to an assembled system. In the linear brick (L8) elements used in this study, an 8 × 8 linear equation system is solved for. The values found for a given node from surrounding elements are then averaged to yield the final interpolated value.

6.5.3 *Reported results*

The elastoplastic finite element model PLAST-PT3 is a displacement formulation, in which the displacements are the unknowns to be solved for. However, the solution requires the calculation of stresses and strains at the quadrature (integration) points within each element, with residual force computations to ensure that the stress state remains within or on the plastic yield surface. Thus a great deal more information about the material state is actually available. In particular, the following parameters are found as a routine part of the calculation at the quadrature points:

(i) the effective stress vector, $\underline{\sigma}'$;

(ii) the strain vector, $\underline{\varepsilon}$;

(iii) the effective plastic strain, $\bar{\varepsilon}_p$;

(iv) the Lode angle, θ.

Furthermore, the reactions $\underline{R}$ are found at the fixed nodes.

Moreover, the magnitude and directions of the principal stress vectors can be calculated from the stress tensor, and the volumetric strain is found from the strain tensor simply as

$$\varepsilon_v = \varepsilon_{ii} = \mathrm{Tr}(\varepsilon_{ij}) \tag{6.98}$$

The parameters listed above are found by default at the quadrature points, but they can be interpolated to the nodes (which coincide with the finite difference block centres in the coupled model) using the smoothing techniques discussed in the previous section.

The Lode angle, θ, based on the parameter described by Lode (1926), is a measure of the value of the intermediate principal stress relative to the minor and major principal stresses (Naylor, 1978), i.e., as the intermediate stress varies linearly from $\sigma_2 = \sigma_3$ to $\sigma_2 = \sigma_1$, θ varies linearly from +30° to −30°. The Lode angle is found explicitly from the expression:

$$\tan\theta = \frac{\sigma_1 - 2\sigma_2 + \sigma_3}{\sqrt{3}(\sigma_1 - \sigma_3)} \tag{6.99}$$

It can also be found in terms of the deviatoric stress invariants (see Appendix F), which is computationally much simpler than finding the principal stresses:

$$\sin 3\theta = \frac{-3\sqrt{3}}{2} \frac{J_3'}{J_2'^{\frac{3}{2}}} \tag{6.100}$$

The Lode angle is used as a stress invariant when describing the yield surface, but is also useful when tracking the stress state of a point in the material. The change in θ from +30° to −30° is usually an indication of principal stress rotation, which to determine otherwise would require the extensive calculations of the principal stresses.

A stress ratio, K_o, is defined in two-dimensional stress analysis as the ratio of horizontal to vertical stresses, viz.,

$$K_o = \frac{\sigma_H}{\sigma_V} \tag{6.101}$$

This is a useful definition for the initial stress state as the principal stress directions will coincide with the x and z axes. A three-dimensional version of K_o can be defined as

$$K_o = \frac{\sigma_{H_{max}}}{\sigma_V} \tag{6.102}$$

where $\sigma_{H_{max}} = \max(\sigma_{H_1}, \sigma_{H_2})$, and is the *smaller* of the two values, given the tension-positive convention of continuum mechanics, so long as a compressive stress field prevails. If the system goes into tension, the statement about the relative size of the two horizontal stresses is no longer true.

A hydraulic fracture or parting in an oil sand is a tensile failure of the material. The parting plane will orient itself such that it is normal to the least principal stress.

Now, assuming that the system remains in a compressive stress regime,

(i) $K_o < 1$ indicates $|\sigma_{H_{max}}| < |\sigma_V|$, i.e., that a vertical parting is anticipated.

(ii) $K_o > 1$ indicates $|\sigma_V| < |\sigma_{H_{max}}|$, i.e., that a horizontal parting is possible, provided that $|\sigma_V| < |\sigma_{H_{min}}|$ is also true.

If this information is combined with the Lode angle and knowledge of the initial stress state, one can determine whether the intermediate stress is the vertical stress or one of the horizontal stresses. If the horizontal stresses are initially equal and the minor principal stresses ($\sigma_2 = \sigma_3$) then the Lode angle will be +30° and K_o will have a value less than one. If the stress state changes due to the accumulation of axisymmetric horizontal loads, then the horizontal stresses will become the major principal stresses ($\sigma_2 = \sigma_1$), the Lode angle will become −30°, and K_o will have a value greater than one. By observing the Lode angle, one has verified that the vertical stress has become the minor principal stress. These statements can be summarized as:

If the horizontal stresses are equal, then, if

(i) $K_o < 1$ and $\theta = +30°$, then $\sigma_{H_1} = \sigma_{H_2} = \sigma_2 = \sigma_3$ and a vertical hydraulic parting is most likely;

(ii) $K_o > 1$ and $\theta = -30°$, then $\sigma_{H_1} = \sigma_{H_2} = \sigma_2 = \sigma_1$ and a horizontal hydraulic parting is most likely.

This presentation of data is useful because it avoids one having to evaluate the principal stresses directly, and because one is normally only interested in whether the parting will be horizontal or vertical, for the purpose of steam stimulation operations. It is predicated on two assumptions, not always true, that

(i) the horizontal stresses are equal; and

(ii) the vertical and horizontal stress directions coincide with the principal stresses.

If it is suspected that the principal stresses will not be aligned with horizontal and vertical planes, then it will be necessary to evaluate them explicitly to be certain of their relative values and directions. If only the values of the principal stresses are required, then they can be found by solving the characteristic polynomial formed by the stress invariants (e.g., Mase, 1970)

$$\sigma^3 - I_1\sigma^2 - I_2\sigma - I_3 = 0 \tag{6.103}$$

This cubic equation is easily solved using Cardan's method (e.g., Bronstein and Semendyayev, 1985). If the directions are required, then an eigenvector calculation must be made. This is most easily done using one of the EISPACK routines available from IMSL, or in the case of this work, the NSWC mathematical library (Morris, 1990). The problem now arises of presenting this information without confusing the reader. There are three direction components and one magnitude for each principal stress. In two dimensions, this information is best presented as an arrow with the correct orientation on the paper and a length proportional to the stress magnitude. In three dimensions the task is much more difficult without the latest specialized software and hardware. This work presents a method not seen previously for providing a quick look interpretation of the principal stresses. In this work, we are particularly interested in which principal stress is closest to the vertical, and the tilt with respect to the vertical of the defining plane of this principal stress.

To find the principal stress closest to the vertical, take the dot product of the first principal stress eigenvector $\underline{A} = (a_x, a_y, a_z)$ with the unit vector in the z-direction, $\hat{k} = (0, 0, 1)$:

$$\underline{A} \cdot \hat{k} = |A| \left|\hat{k}\right| \cos\theta_{A,\hat{k}} \tag{6.104}$$

where θ is the angle made between the two vectors. Whence

$$\theta = \cos^{-1} \frac{a_z}{\pm\sqrt{a_x^2 + a_y^2 + a_z^2}} \tag{6.105}$$

Note that this expression does not guarantee a unique direction for the eigenvector because of the two solutions to the square root. This calculation is repeated with the two other eigenvectors $\underline{B}$ and $\underline{C}$. The value of θ closest to zero is the one closest to the z-direction. It is better to find the value of $\cos\theta$ closest to one, to avoid evaluation of the cosine. This calculation can be repeated with the x-direction, after which the direction of the remaining eigenvector can be inferred because it is normal to the plane specified by the previous two eigenvectors in three dimensions. The principal stress information is then provided as the values of each of the principal stresses, followed by two sets of ordered pairs of numbers:

1. the principal stress number closest to the z-direction, and the cosine of the angle between it and the z-direction; and
2. the principal stress number closest to the x-direction, and the cosine of the angle between it and the x-direction.

Then the numerical output can be scanned quickly to see which direction is closest to which principal stress, and statements about probable parting planes can be made quickly. A cautionary note should be made about the eigenvector calculations — it is possible to generate the conjugate direction rather than the true direction from the eigenvector routines. In other words, there is no guarantee of correctness of sign. It is therefore necessary to monitor changes in the principal stresses and to interpret sudden reversals of direction carefully.

6.6 *Coupling schemes between the models*

Various options exist for coupling the solid and fluid models. For each variation, it is stipulated that the block centres in the finite difference model coincide with the mesh nodes in the finite element model. The options are broadly as follows:

1. The models are explicitly coupled at the level of the Newton-Raphson linearization steps of the fluid model, where the increments of pressure, temperature, and saturation, are applied as loads to the material. The material response as ϵ_v and $\mathbf{v}_r$ is then supplied to the fluid model for solving the next Newton-Raphson iteration. The problem has converged to a solution when the fluid model converges. A stricter convergence requires the solid response to be found to

this last (converged) iteration, and the fluid model rechecked for convergence. Two different approaches are possible in this method:

a) the solid terms are evaluated directly and mostly added to the right-hand-side vector in the equations; and,

b) a pore compressibility term is found which replaces the pore compressibility used in the original finite difference model, while the remaining solid terms are added to the right-hand-side vector in the equations.

The second approach contains a greater degree of implicitness in the solution, but might induce too great changes.

2. The models can be run in a staggered mode, where the solid model provides information to the fluid model with a lag of one time step. Thus the fluid model incorporates solid parameters which reflect changes to the distribution of pressure, temperature, and saturation at the previous time. The solution can also be found by incorporating a pore compressibility derivative with respect to pressure, which is found from the preceding time step, and is used to predict the pore compressibility in the current time step.
3. The models are run so there is no feedback to the fluid model, and one sees the uncoupled response of the solid model to the changes in pressure, temperature, and saturations.

6.6.1 *Explicit coupling in Newton-Raphson iterations*

The finite element model is explicitly fully coupled to the finite difference thermal fluid model. This is achieved by stepping through the models at the Newton-Raphson iteration level of the finite difference model. First, a satisfactory solution is found for the finite difference model, without calling for the solid solution. Thus the finite difference model will select an appropriate time step which provides convergence. At that time, the successful time step is re-run, but now, after each successive Newton-Raphson iteration, the increments of pressure, temperature, and bulk density are applied as loads to the finite element model. The resultant solid velocity $\mathbf{v}_r$ and the volumetric strain $\varepsilon_v = \mathrm{Tr}(\varepsilon_{ij})$ are returned to the finite difference model to generate the solid coupling terms. Thus the solid model lags behind the fluid model by one Newton-Raphson iteration. Strict coupled convergence is considered to have been achieved when the fluid model has converged on two successive Newton-Raphson iterations, confirming that the last increment of load had a negligible effect on the material. It is also permissible to pass over the confirmatory iteration. If coupled convergence is not achieved within the allotted number of Newton-Raphson iteration steps, the time step size is reduced

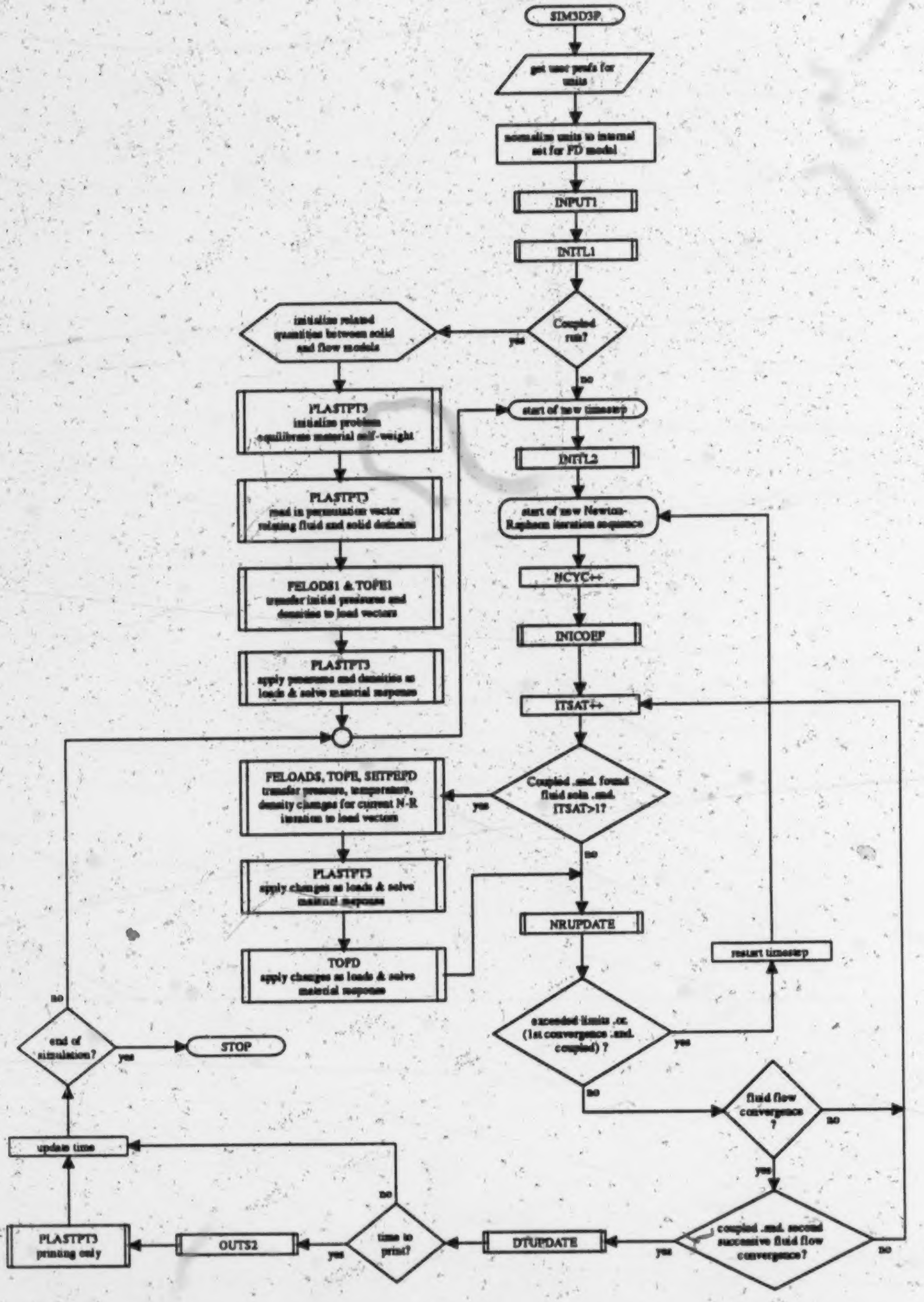

SIM3D3P
get user prefs for units
normalize units to internal set for FD model
INPUT1
INITL1
Coupled run?
yes
no
initialize related quantities between solid and flow models
PLASTPT3 initialize problem equilibrate material self-weight
PLASTPT3 read in permutation vector relating fluid and solid domains
FELODS1 & TOFE1 transfer initial pressures and densities to load vectors
PLASTPT3 apply pressures and densities as loads & solve material response
start of new timestep
INITL2
start of new Newton-Raphson iteration sequence
NCYC++
INICOEF
ITSAT++
Coupled .and. found fluid soln .and. ITSAT>1?
yes
no
FELOADS, TOFE, SETPEFD transfer pressure, temperature, density changes for current N-R iteration to load vectors
PLASTPT3 apply changes as loads & solve material response
TOFD apply changes as loads & solve material response
NRUPDATE
exceeded limits .or. (1st convergence .and. coupled) ?
yes
no
restart timestep
fluid flow convergence ?
no
yes
coupled .and. second successive fluid flow convergence?
yes
no
DTUPDATE
time to print?
yes
no
OUTS2
PLASTPT3 printing only
update time
end of simulation?
yes
no
STOP

Figure 6–3

Coupling models at level of Newton-Raphson iterations.

and the problem re-run for the new, smaller, time step. Figure 6–3 is a flow chart illustrating the procedural logic of the coupled problem (see the next section for a description of the subroutines used in this program).

Due to the irreversible nature of accumulated plastic strain, it is important to ensure that the loads applied each (fluid) Newton-Raphson iteration step do not change direction significantly. The Newton-Raphson linearisation scheme is a tangent search method, and it is possible for the true (fluid) solution to be overshot so that subsequent iterations push the incremental solution vectors in the other direction. Overshoot of the solution can be controlled by retarding the size of the incremental solution vectors—achieved using under-relaxation. If overshoot is recognized as a problem, under-relaxation can be applied during a specified number of the early iterations.

Several options are available to adjust the method. One is to hold the solid velocity terms at the values from the previous time step, another is not to update them after the first n Newton-Raphson iterations.

A bigger variation on this method permits the solid response to be more implicitly coupled to the fluid model, and, in theory, be a more stable formulation. The approach here is to take the terms containing the time derivative of porosity, and treat them as in the classic finite difference reservoir simulator formulations, viz., with a pore compressibility term. Thus one can write, for the first order approximation of the pore compressibility relation,

$$\begin{aligned}\frac{\partial \phi}{\partial t} &= \frac{\partial \phi}{\partial p}\frac{\partial p}{\partial t} \\ &= \frac{\partial}{\partial p}\left[\phi_0(1 + c_\phi \Delta p)\right]\frac{\partial p}{\partial t} \\ &= (\phi_0 c_\phi + \Delta p)\,\frac{\partial c_\phi}{\partial p}\frac{\partial p}{\partial t} \end{aligned} \tag{6.106}$$

or, for the full pore compressibility relation,

$$\begin{aligned}\frac{\partial \phi}{\partial t} &= \frac{\partial \phi}{\partial p}\frac{\partial p}{\partial t} \\ &= \frac{\partial}{\partial p}\left[\phi_0 \exp^{(c_\phi \Delta p)}\right]\frac{\partial p}{\partial t} \\ &= \phi_0\left[c_\phi \exp^{(c_\phi \Delta p)} + \Delta p \frac{\partial c_\phi}{\partial p}\exp^{(c_\phi \Delta p)}\right]\frac{\partial p}{\partial t} \\ &= \phi\left(c_\phi + \Delta p\frac{\partial c_\phi}{\partial p}\right)\frac{\partial p}{\partial t} \end{aligned} \tag{6.107}$$

Note the difference between the two results. The above analysis put the pore compressibility as a function of pressure. This refutes the assumption that pore compressibility is a constant when integrating the equation defining it. This inconsistency can be resolved by considering pressure dependency when integrating. A relation with pressure can be assumed, lacking any experimental information. Assume, then, a linear function of pressure for the pore compressibility:

$$c_\phi = ap \tag{6.108}$$

Integrating,

$$\int_{p_0}^{p_1} ap\,dp = \int_{\phi_0}^{\phi_1} \frac{d\phi}{\phi} \tag{6.109}$$

whence,

$$\frac{a}{2}\left(p_1^2 - p_0^2\right) = \ln\frac{\phi_1}{\phi_0} \tag{6.110}$$

and,

$$\phi = \phi_0 \exp^{(a/2)(p_1^2 - p_0^2)} \tag{6.111}$$

from which we extract a value of a and then c_ϕ, using the relation between volumetric strain, thermal expansion, and porosity change. At present, this has not been implemented in the model.

When using the pore compressibility coupling formulation, the changing values of pore compressibility over each time step require that the porosity be found in an incremental manner from time step to time step, rather than from some arbitrary base value, i.e.,

$$\phi^{n+1} = \phi^n \exp^{(c_\phi^{n+1}\Delta^n p)} \tag{6.112}$$

where $\Delta^n p = p^{n+1} - p^n$. Pore compressibility is found from $\Delta^n \varepsilon_v$ and $\Delta^n T$, and therefore represents an average value for the time step. The derivative

$$\frac{\partial\phi}{\partial p} = \phi^{n+1}\left(c_\phi^{n+1} + \Delta^n p \left.\frac{\partial c_\phi}{\partial p}\right|_{n+1}\right) \tag{6.113}$$

is approximated as

$$\begin{aligned}\frac{\partial\phi}{\partial p} &\approx \phi^{n+1}\left(c_\phi^{n+1} + \Delta^n p\frac{\Delta^n c_\phi}{\Delta^n p}\right)\\ &= \phi^{n+1}\left(2c_\phi^{n+1} - c_\phi^n\right)\end{aligned} \tag{6.114}$$

When the fluid solution is found for the first sequence of the time step, the previously found values of volumetric strain are projected using the approximation:

$$\begin{aligned}\varepsilon_v^{n+1} &= \varepsilon_v^n + \frac{\partial \varepsilon_v}{\partial t} \\ &\approx \varepsilon_v^n + \frac{\Delta^{n-1}\varepsilon_v}{\Delta^{n-1}t}\Delta^n t \\ &= \varepsilon_v^n + \frac{\varepsilon_v^n - \varepsilon_v^{n-1}}{t^n - t^{n-1}}(t^{n+1} - t^n) \qquad (6.115)\end{aligned}$$

This approximation ensures that a smooth change in ε_v is felt in the initial solution for pressure. An approximation using a user-input value of pore compressibility is made for the first iteration of the first time step.

6.6.2 *Staggered coupling (solid solution lagging)*

A simplified solution scheme is proposed where the solid solution is made to lag behind the heat and fluid solution by one time step. Its principal advantage is that only one call need be made to the solid program per time step. The values of $\Delta\varepsilon_v$ are then always at the previous time level. The influence of the solid solution can be extended to the $n+1$ time level by calculating a new pore compressibility from the time-lagged solution for volumetric strain, and then forming a numerical derivative

$$\frac{\partial c_\phi}{\partial p} = \frac{c_\phi^n - c_\phi^{n-1}}{p^n - p^{n-1}} \qquad (6.116)$$

This derivative can then be used to make pore compressibility a pressure-dependent variable when solving for pressures, saturations, and temperatures at the $n+1$ time level.

6.6.3 *No feedback to fluid model*

The simplest level of coupling is to remove any feedback from the solid model into the finite difference thermal fluid model. This form of analysis is useful for gauging the extent and degree of plastic deformation as a result of hot fluid injection. While they do not influence the fluid solution, the results permit conclusions to be drawn and inferences made about the likely outcome of hot fluid injection into such a material.

6.7 *Description of subroutines in coupled finite-difference and finite element models*

The subroutines constituting the implementation of the two models are described here. The fluid model is a proprietary model and full source code may not be given. However, adequate description is given considering the equations and the purpose of the routines.

6.7.1 *Solid model routines*

The following subroutines describe the implementation of the finite element elastoplastic solid model. The file in which the subroutine appears follows the subroutine name in brackets.

ALGOR (algor.f)

This subroutine sets the equation resolution index, which governs whether the element stiffness matrix is updated.

BMATPS (bmatps.f)

This subroutine evaluates the strain-displacement matrix, B.

CHCKRST (restarts.f)

This subroutine checks to see whether the restart data is consistent with the main data file.

CHECK0 (check0.f)

This subroutine checks the initial control data for valid ranges.

CHECK1 (check1.f)

This subroutine checks the main control data for valid ranges.

CHECK2 (check2.f)

This subroutine checks the remainder of the input data.

CONVER (conver.f)

This subroutine checks for convergence of the iteration process.

DBE (dbe.f)

This subroutine returns the product **DB**.

DIMEN (dimen.f)

This subroutine presets variables associated with dynamic dimensioning.

DIR3D (pstress.f)

This subroutine provides abbreviated directional information from the eigenvectors eigv(), with respect to each of the z- and x-directions in turn.

ECHO (echo.f)

If data errors have been detected by subroutines CHECK1, CHECK2, or CHECK3, this subroutine reads and writes out the remaining data cards.

ELGLOB (faceinfo.f)

This subroutine returns the local (element) node number from the global node numbers.

FACES (faceinfo.f)

This subroutine returns the element face number defined by the local nodes supplied.

FECHEK (fdcpl.f)

This subroutine equates PLAST-PT3 convergence to a global variable.

FEFDINIT (fdcpl.f)

This subroutine initialises related parameters common to both FE and FD modules in program.

FEREADY (fdcpl.f)

This subroutine initialises applied loads to zero and to revert values if required.

FLOWPL (flowpl.f)

This subroutine evaluates the plastic stress-strain constitutive D vector.

front (front.f)

This subroutine undertakes equation solution by the frontal method.

GAUSSQ (gaussq.f)

This subroutine sets up the gauss-legendre integration constants for 3-D L8 and L6 elements.

INCREM (increm.f)

This subroutine increments the applied loading.

INIPLST (plinit.f)

This subroutine initialises variables for PLAST-PT3.

INITST (initst.f)

This subroutine calculates the equivalent nodal forces for the cumulative initial stresses.

INPUT (input.f)

This subroutine accepts most of the input data, plus any initial stresses.

INVAR (invar.f)

This subroutine evaluates the stress invariants and the current value of the yield function.

JACOB3 (jacob3.f)

This subroutine evaluates the Jacobian matrix and the Cartesian shape function derivatives.

LCLAVG (lclavg.f)

This subroutine prepares and calls for the solution of nodal averages using Galerkin method — local averaging only.

LINEAR (linear.f)

This subroutine evaluates stresses and strains assuming linear elastic behaviour.

LOADPS1 (loadps1.f)

This subroutine evaluates the consistent nodal forces for each element — initial loadings.

LOADPS2 (loadps2.f)

This subroutine evaluates the consistent nodal forces for each element — incremental loadings.

MODPS (modps.f)

This subroutine evaluates the elastic constitutive **D** matrix.

NEWLOAD (newload.f)

This subroutine sets up the values of change in pore pressure, temperature, and bulk density, for a given instant in time.

NODEXY (nodexy.f)

This subroutine interpolates the mid-side nodes of straight sides of elements and the central node of 9 noded elements.

OUTPUT (output.f)

This subroutine outputs displacements, reactions and stresses.

PLASTPT3 (plastpt3.f)

This is the main controlling routine of the finite element program.

PRSTIN (restarts.f)

This subroutine reads the current variables from the restart file according to the appropriate time step number.

PRSTOUT (restarts.f)

This subroutine prints the current variables to the restart file.

DIR3D (pstrclc.f)

This subroutine provides abbreviated directional information from the eigenvectors eigv(), with respect to each of the z- and x-directions in turn.

PSTPREP (pstrspr.f)

This subroutine returns the principal stresses and their vectors.

RESIDU (residu.f)

This subroutine reduces the stresses to the yield surface and evaluates the equivalent nodal forces.

SFR3 (sfr3.f)

This subroutine evaluates shape functions and their derivatives for L8 linear brick elements and L6 linear triangular prism elements (both 3-D).

SJACOB3 (sjacob3.f)

This subroutine evaluates the surface multiplier for Gauss integration, equivalent to the Jacobian for 3-D integration.

STIFFP (stiffp.f)

This subroutine evaluates the stiffness matrix for each element in turn.

TOFD (fdcpl.f)

This subroutine converts parameters from the FE model to ordering that the FD model uses. It passes the latest values in volumetric strain and displacement velocities.

TOFE (fdcpl.f)

This subroutine converts parameters from the FD model to ordering that the FE model uses. It passes the latest changes in temperature, pressure, and density.

TOFE1 (fdcpl.f)

This subroutine converts parameters from FD model to ordering that the FE model uses. It passes the initial values in temperature, pressure, and density. Note that an initial density was used for the initial stress calculations, so that this is altered for use with PLAST-PT3.

V2MAX (loadps2.f)

This subroutine finds the maximum absolute value of a 2-D array.

VITABLE (newload.f)

Routine to write out the reservoir parameters for perusal (integer version).

VTABLE (newload.f)

Routine to write out the reservoir parameters for perusal (real version).

YIELDF (yieldf.f)

This subroutine evaluates the flow (rule) vector for three dimensions.

ZERO (zero.f)

This subroutine initialises various arrays to zero.

ZEROEV (fdcpl.f)

This subroutine zeros the volumetric strain vector before applying loads from injection or production.

A small number of function subroutines are used in the finite element model:

INSET (faceinfo.f)

This function tells whether the input vector l_n is equal to the set $\{l1, l2, l3, l4\}$.

INSUBD (pstrspr.f)

This function checks whether the point IPOIN is in the fluid flow subdomain; location ILOC is the permutation vector index.

6.7.2 *Fluid model routines*

The following subroutines describe the implementation of the finite difference thermal fluid model. The file in which the subroutine appears follows the subroutine name in brackets.

CALCNEW (couplings.f)

This subroutine calculates the new terms used in the fluid model when it is coupled to the solid model.

CALCOEF (calcoef.f)

This subroutine prepares the coefficient matrix and constant vector for the primary variable solutions.

CLCTRANS (couplings.f)

This subroutine prepares the constant part of the transmissibility term using the permeability modifications.

CPLOUT (couplings.f)

This subroutine prints out coupling parameters for debugging.

DOMOL1 (smfa_units.f)

This subroutine makes a molar quantity out of a mass-based one.

DTUPDATE (updates.f)

This subroutine updates variables preparatory to performing a new time step.

FDFESPS (fdfesps.f)

This is the main program header, and invokes the type of simulation requested by the user.

FELOADS (couplings.f)

This subroutine prepares the changes in pressure, temperature and bulk density to be used as loads in the finite element model.

FELODS1 (couplings.f)

This subroutine prepares the first set of changes in pressure and bulk density to be used as loads in the finite element model. The initial temperature field is the departure point for temperature changes, so it is not applied as a separate load during initialization.

FEPOUT (couplings.f)

This subroutine prints out pore compressibility data.

FEPRJCT (couplings.f)

This subroutine projects ϵ_v and $\mathbf{v}_r$ values from the previous time step.

FEPROP (couplings.f)

This subroutine calculates the new terms used as basic properties (porosity, absolute permeability).

GOPLAST (sim3d3p.f)

This subroutine executes PLAST-PT3 for latest Newton-Raphson iteration.

INICOEF (initial.f)

This subroutine calculates the heat loss to the overburden.

INITL1 (initial.f)

This is a parameter initialisation routine run once at the start of the simulation.

INITL2 (initial.f)

This is a parameter initialisation routine run once at the start of a new time step.

INPUT1 (ins.f)

This routine reads in the main data for the fluid model.

INPUT2 (ins.f)

This routine reads in well data for the fluid model.

INUSER (smfa_units.f)

This routine reads in user defined input units and sets up the vector of constants to convert them to internal (SI) units.

IOUSER (smfa_units.f)

This routine controls the units routines and prints the final table.

NRUPDATE (updates.f)

This subroutine does NR iteration check, followed by updating or reverting for a repeat series of cycles.

OUTCUP (outs.f)

This subroutine outputs coupled data at the same times as does OUTS2.

OUTS1 (outs.f)

This subroutine outputs the initial report at the start of the simulation.

OUTS2 (outs.f)

This subroutine outputs reports at user defined times.

OUTUSER (smfa_units.f)

This routine reads in user defined output units and sets up the vector of constants to convert them to internal (SI) units.

PRJCTN (sim3d3p.f)

This routine controls the prediction of approximations to solid parameters at start of a time step.

SETFEFD (couplings.f)

This subroutine sets shared values for the solid and fluid models.

SIM3D3P (sim3d3p.f)

This subroutine starts the entire simulation.

SIMCTL (sim3d3p.f)

This subroutine controls the entire simulation for every time step.

SMFUNITS (smfa_units.f)

This subroutine converts unit scales from pure S.I. to those originally used in SIM3D3P.

STRTPLST (sim3d3p.f)

This subroutine starts PLAST-PT3 by initialising all stresses and reactions.

Chapter VII

PREPARATION AND ANALYSIS OF DATA: PRE- AND POST-PROCESSING

A basic problem with the interpretation of data is the appropriate presentation for analysis. Arranging the data for convenient access and graphing is of the utmost importance, and where possible, the use of animation permits the overview of a sequence of events in a short time. The presentation of three-dimensional data has the additional problem of portraying a third dimension on a planar display. Several tools have been developed to aid in the preparation and analysis of data from the numerical models described in this thesis.

(i) A grid generation program, `grid`, assists in describing the three-dimensional mesh for the finite element program, giving a convenient method for describing element connectivities (incidence matrices), nodal fixities (prescribed displacements), and the permutation vector describing the position of the fluid flow domain in the global mesh.

(ii) An element connectivity program, `create_conns`, processes the three-dimensional finite element grid to produce polygon vertex sets which can be input into a three-dimensional display program such as Mathematica.

(iii) A program, `dvar`, processes pressure and temperature data from the fluid flow model preparatory to input to the solid model as loads.

(iv) A post-processing program for the main output file from the finite element model, `postp`, extracts user-specified data from otherwise unwieldy files (frequently in excess of 50 Mbytes).

(v) A program, `conprep`, processes data summarized from either the solid model by `postp` or directly from one of the output files from the fluid flow model. The data is stored to the public domain HDF file format for later use.

(vi) A program, `conslice`, takes three-dimensional HDF data and extracts a slice or plane of values, saving either to an ASCII table or to another HDF file, or both.

(vii) Two public domain programs, `fp2hdf` and `Image3`, are used to generate raster images of the planar data from the files made by `conslice`, and then to create animated sequences of the raster images.

7.1 *Using the pre-processing routines*

The pre-processing program, **grid**, is required to generate the finite element mesh for PLAST-PT3. If the flow model is run separately from the solid model, then the output file IPLAST from the flow model must be pre-processed by **dvar** to provide suitable input to PLAST-PT3. The preparation of data for and the use of these programs is discussed below.

7.2 *grid*

grid is a FORTRAN77 program for finite element generation. The program is compiled by Sun `f77` using the following command:

```
f77 -u -g -f68881 -o grid grid.f
```

The grid program can generate 3-dimensional rectilinear meshes using 8-noded brick elements. The output information includes the element connectivity matrix (relating element nodes to global node numbering) complete with identification of element material, the coordinates of the nodes, a table of nodal fixities (constraints on displacements in any combination of the x-, y-, and z-directions), and the permutation vector identifying a specified subdomain where the flow model operates. Totals in all categories are also provided as comments. The data can be used directly with PLAST-PT3 as input.

The **grid** program is simple to run, using only standard input and output:

```
grid < grid_input > grid_output
```

A typical input file for grid generation is as follows:

```
# comment character
# mesh generation
#
# x, y, z number of nodes
  8  8  14
#
# coordinates of first node
#  x       y       z
  0.0     0.0   -535.4
#
# x-lengths from the start position
40.0  40.0  4.0   10.0  20.0  40.0  100.0
#
# y-lengths from the start position
40.0  40.0  4.0   10.0  20.0  40.0  100.0
```

```
#
# z-lengths from the start position
40.0  20.0  10.0  4.0   3.7   3.7   4.0   10.0  20.0  40.0  80.0  150.0 150.0
#
# plane in which nodes are numbered (x,y,z) == (1,2,3)
   3
#
# start node numbering in which direction (x,y,z) == (1,2,3)
   1
#
# fixities are additive: only worry about those acting on whole face!
# fixities can exist on all six outside faces of domain.  Note the
# fixity code (bbb) where b = 0 or 1, and corresponds to positions (xyz).
#
# x+ plane fixities
100   0.0   0.0   0.0
# x- plane fixities
100   0.0   0.0   0.0
#
# y+ plane fixities
010   0.0   0.0   0.0
# y- plane fixities
010   0.0   0.0   0.0
#
# z+ plane fixities (top surface is free surface)
000   0.0   0.0   0.0
# z- plane fixities
001   0.0   0.0   0.0
#
# number of supplementary material types to associate with elements
# (i.e., above default number 1, which the main body takes)
   2
# numbering is by element in the same (x,y,z) directions as for the nodes
# note that the numbers can overwrite previous definitions
# matno  xmin, xmax, ymin, ymax, zmin, zmax
   2      1     7     1     7     7     13
   3      1     7     1     7     1     4
#
# subdomain origin in global domain
# xmin, ymin, zmin
   1     1     7
# extent of subdomain, relative to global domain
# note that we chose z +ve upwards, opposite to typical reservoir
# simulator for which the permutation vector is intended
# snx,  sny,  snz
   3     3     -3
```

7.3 *dvar*

When running the fluid flow and solid models separately, the output from SIM3D3P can be processed to give loads that PLAST-PT3 can read as input. The condensed output file IPLAST generated by SIM3D3P can be processed by **dvar** to give the changes in pore pressure and temperature for each time step, which are then used as loads. Note that at the zeroth time step, there is an initial pore pressure loading to be applied, while there is a zero temperature field to be imposed (initial temperature). The units are also changed to kPa and °C, which are required by PLAST-PT3. dvar can also generate tables of data with coordinate information, information which must be supplied in the header file.

dvar takes one file as input: the output file IPLAST from SIM3D3P with a header prepended to it. Only the short header will be described. Two important lines at the end of the header inform the program which data are present in the file, and which data should be processed. **dvar** is simply invoked using standard input and output:

```
dvar < data > dpdt_output
```

A typical header segment is as follows:

```
# start with comment character
#
# PROBLEM: 0: do incremental values between each time step
#          1: do total change since start.
# COLS:    1: sort data into cols with (x,y,z) coordinates
# UCONV:   0: do no units conversion
# UCONV:   1: do default units conversion
# problem, cols, uconv
  0, 0, 1
# start of data from FDFESPS
# Started analysis on Wed Jun 26 13:35:31 1991
#
# all data written out as NX, NY, NZ
      5      5      3
#
#
# read in:
# p   T   So   Sw   Sg  den_b   phi   kfact
  1   2    3    4    5    0      0     0
# write out:
# p   T   So   Sw   Sg  den_b   phi   kfact
  1   2    0    0    0    0      0     0
#
```

7.4 *create_conns*

create_conns is a C program to create (x, y) connectivity lists from the element connectivity matrix. The purpose is to re-describe the original matrix connectivity list in terms of either the element edges (lines) or faces (quadrilaterals). The program produces the minimum set of objects required to draw the domain with the element boundaries, either as a complete set with internal elements, or merely the exterior faces. This information can then be input to a program capable of generating three-dimensional images for fast processing into such an image. The procedure will be described for using Mathematica®, a commercially-available program. **create_conns** requires two input files if only the exterior surface need be rendered, or one if the entire domain will be described. In both cases an input file will be required with information defining the problem and the element connectivity matrix, and is supplied as standard input. The second file, if required, is a list of nodal coordinates and is read in using a file name given in the first data file. The order of the output reflects the order of the elements of the original input file.

The program is normally invoked directly:

```
create_conns < my_input > my_output
```

The main input file takes the following format:

1. **'debug'**

   This optional word indicates that visual feedback is provided as the elements are processed, in the form of a dot every ten elements that are passed through.

2. **data_type**

   data_type = 'line': generate connectivities as lines joining the element vertices to define the element edges.

   data_type = 'poly': generate connectivities as quadrilaterals defining the element faces.

3. **data_type coord_fname xface yface zface xmin xmax ymin ymax zmin zmax**

   Place a '#' at the start of this line if it is desired that the entire domain, including the interior elements, be rendered.

   data_type = 'limits': generate data only for the exterior faces of the domain.

   xface yface zface: these data indicate which faces will be defined in each of the x, y, z directions. One flag is provided for each coordinate:

   xface = 0: negative x-direction,

   xface = 1: positive x-direction,

   xface > 1: both x-directions,

   xmin xmax ymin ymax zmin zmax: these data indicate the limiting values in each of the x, y, z

directions, being the extremes in the negative and positive directions for each coordinate.

`xface yface zface`: these data indicate which faces will be defined in each of the x, y, z directions. One flag is provided for each coordinate:

4. `i mat n1 n2 n3 n4 n5 n6 n7 n8`

   This is the list of element incidence matrices where `i` is the element number, `mat` is the material ID number (superfluous, any integer will do), and `n1` to `n8` are the global defining node numbers for the element. There will be as many lines of data as there are elements constituting the domain.

If the file of nodal coordinate data is required, it takes the simple form of :

1. `x_coord y_coord z_coord`

where the nodes are implicitly numbered sequentially.

7.4.1 *Using Mathematica(R) with output from create_conns*

Mathematica has powerful drawing capabilities, and can easily render a three-dimensional object that is adequately defined. The object can then be viewed from any desired angle and distance using the interactive tools available in the application, and can be exported as an EPS file for annotation and embellishment before using it as an illustration. The output from `create_cons` is used with the same file of nodal coordinates mentioned in the previous section. The following set of Mathematica command lines will input the required data and draw the solid body:

```
xyz = ReadList ["nodal_coords", {Number,Number,Number}];
connpoly = ReadList ["test_ext_polys",
{Number,Number,Number,Number}];
graphpoly = Table[ Polygon[{xyz[[ connpoly[[i,1]] ]],
xyz[[ connpoly[[i,2]] ]], xyz[[ connpoly[[i,3]] ]],
xyz[[ connpoly[[i,4]] ]] } ], {i, Length[connpoly]}];
polydiagram = Graphics3D[ {graphpoly} ]
SetOptions[Graphics3D, Shading->False, Boxed->False]
Show[ polydiagram ]
```

7.5 *Using the Post Processing Routines*

The post-processing programs **postp**, **conprep**, **conslice**, and **fp2hdf**, plus the **Image3** program for the Macintosh, are required for the appropriate analysis of output from PLAST-PT3 and SIM3D3P. The preparation of data for and the use of these programs is discussed below.

7.6 *postp*

`postp` is a C program to extract data from the main output file generated by PLAST-PT3. The program is compiled using the following command:

```
gcc -O -g -o postp postp.c
```

The amount and type of data extracted from the main output file is governed by a short header file which is prepended to the main output file. On a Unix system this is easily achieved by the following pipelined command:

```
cat header_file main_data | postp > result_file
```

A typical header file is as follows:

`TYPE METHOD` (integer codes)

`VARIABLE_SOUGHT` (character data, e.g., `displacement`)

`OTHER_CODES` (integer codes, such as: the number of nodes or Gauss points, the node or element numbers, the time step number, ranges to be ignored.)

No comment markers are currently supported. The currently supported options are listed in the following subsections. The marker | is used to denote the binary OR operator.

The following is a list of the routines other than main which are called in the program:

search11()
search12()
search2()
search3()
search4()
search61()
search62()
search71()
search72()

The first digit at the end of the routine name corresponds to the `TYPE` number, and the second digit corresponds to the `METHOD` number. The formats of the data files are now explained for each `TYPE` number.

7.6.1 *TYPE = 1*

Presents displacements or reactions at specified nodes for every time step. The directions of the data depend on the `METHOD` chosen. dx, dy, and dz data are presented for one specified node if `METHOD = 1`. Only one of the three directions is presented if `METHOD = 2`, but for several node numbers specified in ascending order. Example header files are:

```
1 1
displacements|reactions
NODE_NO
1 2
displacements|reactions
x|y|z-direction
NDATA
NODE_NO_1
NODE_NO_2
```

7.6.2 *TYPE = 2*

Presents effective plastic strain $\bar{\epsilon}_p$, volumetric strain ϵ_v, the stress components σ_{xx}, σ_{yy}, or σ_{zz}, or Lode angle θ at selected Gauss points and elements at each time step. Only `METHOD = 2` is permitted. An example header file is:

```
2 1
EPS|Ev|Sxx|Syy|Szz|Lode
NDATA
ELEM_NO_1 GAUSS_PT_1
ELEM_NO_2 GAUSS_PT_2
ELEM_NO_3 GAUSS_PT_3
```

7.6.3 *TYPE = 3*

This type will generate an echo of all the output data for a specified time step. The method is necessary but irrelevant. An example header file is:

```
3 1
TIMESTEP_NO
```

7.6.4 *TYPE = 4*

Presents, for all time steps, all non-zero values of $\bar{\epsilon}_p$ (for `METHOD = 1`) or negative values of the stress ratio K_o (for `METHOD = 2`). A specified number `NRANGE` of groups of elements can be excluded from the report, and must be specified in ascending order. Each group is identified by the range `MIN` to `MAX`. Thus the growth of the sheared zone can be identified, or the development of either vertical or horizontal positive (tensile) stresses. The number of expected Gauss points must be provided (generally eight). An example header file is:

```
4 1|2
NGAUS
NRANGE
MIN[0] MAX[0]
MIN[1] MAX[1]
MIN[2] MAX[2]
MIN[NRANGE] MAX[NRANGE]
```

7.6.5 *TYPE = 5*

Presents, for all time steps, all tensile (positive) stresses. A specified number, `NRANGE`, of groups of elements can be excluded from the report, and must be specified in ascending order. Each group is identified by the range `MIN` to `MAX`. Thus the development of positive (tensile) stresses can be identified. The number of expected Gauss points must be provided (generally eight). The `METHOD` identifier is necessary but irrelevant in this case. An example header file is:

```
5 1
NGAUS
NRANGE
MIN[0] MAX[0]
MIN[1] MAX[1]
MIN[2] MAX[2]
MIN[NRANGE] MAX[NRANGE]
```

7.6.6 *TYPE = 6*

Presents all values of one variable within one time step. Nodal variables such as displacements and reactions are generated for `METHOD = 1`, and values at Gauss points such as $\bar{\epsilon}_p$, ϵ_v, σ_{xx}, σ_{yy}, σ_{zz}, or Lode angle θ for `METHOD = 2`. In the latter case, the number of Gauss points to be reported must be specified after the time step number. Example header files are:

```
6 1
Displacements|Reactions
```

```
TIMESTEP_NO
6 2
EPS|Ev|Sxx|Syy|Szz|Lode
TIMESTEP_NO NGAUSS_PTS
```

7.6.7 *TYPE = 7*

Presents all values of one variable for all time steps up to a specified time step. Nodal variables such as displacements and reactions are generated for `METHOD = 1`, and values at Gauss points such as $\bar{\epsilon}_p$, ϵ_v, σ_{xx}, σ_{yy}, σ_{zz}, or Lode angle θ for `METHOD = 2`. In the latter case, the number of Gauss points to be reported must be specified after the time step number. Example header files are:

```
7 1
Displacements|Reactions
TIMESTEP_NO
7 2
EPS|Ev|Sxx|Syy|Szz|Lode
TIMESTEP_NO NGAUSS_PTS
```

7.7 *conprep*

conprep is a FORTRAN77 program to create an HDF (hierarchical data format) file from either the PLAST-PT3 data, via the **postp** program, or from the IPLAST or SPLAST output files from SIM3D3P, the reservoir simulator. **conprep** is a multi-purpose program:

(a) it will generate a 3D HDF file from the output created from **postp** (typically `TYPE` = 7 and `METHOD` = 1 or 2) for a specified subdomain, and for all time steps;

(b) it will generate a 3D HDF file from the output file IPLAST created from SIM3D3P, for a specified subdomain, and for all time steps. The data presented can be in terms of the actual values, changes over a time step, or the cumulative change since the start of the time steps;

(c) preparation of tables of data from **postp** output. The program can generate data according to several options:

  (i) a table of nodal data with *i,j,k* coordinates;

  (ii) a table of nodal data with *x,y,z* coordinates;

  (iii) a table of Gauss point data with *x,y,z* coordinates;

The program will only process one value of data in space, and when considering displacements or reactions, the x, y, or z direction of the variable must be selected.

File	*Subroutine*
conprep.f	conprep coninit
datmat.f	ndatmat edatmat
ioutils.f	getcmmt echocmmt getstring
odats.f	nodals elems loctable
ptnodl.f	ptnodl pstnodl
sdsprep.f	findMaxMin3 sdsprep
setup.f	setgpos setgeom getcods
sfr3.f	sfr3

Table 7–1

Source files and routines constituting CONPREP

Data can be extracted on a plane or along a line, and then tabulated. 3D data can be extracted only when creating the HDF file. The hierarchical data format is a versatile binary storage scheme for scientific data placed in the public domain by the National Center for Supercomputer Applications, University of Illinois at Urbana-Champaign, IL, U.S.A, commonly referred to as the NCSA. A group of floating point data, called a scientific data set (SDS), is combined as an HDF file, to which more SDSs can be appended. This is used to add data for each time step from a simulation. Various visualisation tools are available which require HDF files as input. The format is also convenient and compact for general storage and extractions. The details of the HDF support are available by anonymous ftp from the Internet archive site zaphod.ncsa.uiuc.edu [128.174.20.50].

The table of data generated by `conprep` is a matrix suitable for input into contouring programs which cannot resolve data in a simple (x, y, z) column format, which is also produced.

The program consists of the files and subroutines listed in Table 7–1. The program can be easily compiled on a Unix system by typing `make prep` in the presence of the source files and the supplied `makefile`. The program is run two slightly different ways, depending on whether the output being processed comes from PLAST-PT3 or SIM3D3P. Both cases require a shell script to start the process.

7.7.1 *Scripts MKSDS and MKSDSRUN: for* PLAST-PT3 *output*

This script is run using the command:

```
mksds postp.hdr plast.out conprep1.hdr conprep2.hdr coords grid rows.cols SDSfile
```

The script parameters are explained below.

(i) `postp.hdr` is the header file for the `postp` program (`TYPE` = 7 and `METHOD` = 1 or 2).

(ii) `plast.out` is the main data file output from PLAST-PT3.

(iii) `conprep1.hdr` is the first part of the control data for `conprep`. The type of information depends on whether one is extracting nodal or Gauss point (element) data.

(iv) `conprep2.hdr` contains data for labelling and initialising the HDF file.

(v) `coords` is a file of nodal coordinate data in the form `NODE_NO, X, Y, Z`.

(vi) `grid` is a file containing different connectivity data, depending on whether one has nodal or element data.

  (a) *Nodal data.* The table is a connectivity table relating (i, j, k) locations to node numbers. Node numbers are read in such that the left-hand index varies fastest, i.e., i, then j, and lastly k.

  (b) *Element data.* Two sets of data are required. First, a connectivity table relating (i, j, k) numbering of elements to element number, read in in the same manner as for node numbers. Second, an incidence matrix defining the local node number in each element in terms of the global node numbering, in the form `ELEM_NO, NODE_1, ..., NODE_8`.

(vii) `rows.cols` is a short file containing the number of rows and columns of numeric data output from `postp` in each time step.

(viii) `SDSfile` is the name of the SDS file to be created and written to.

If a constant PLAST-PT3 output file is used, and `postp` and `conprep` header files have been set up, a simpler script can be used:

```
mksdsrun elem|node xx
```

where `xx` is the data of interest (e.g., K_o, displacements, etc.) and either `NODE` or `ELEM` is specified. This script simply invokes the other script, `mksds`.

7.7.2 *Preparing the header files for conprep*

It is already assumed that the main PLAST-PT3 output file has been postprocessed using `postp`, to generate single value data for all points and time steps (using `TYPE` = 7 and `METHOD` = 1 or 2). The examples are divided into nodal and element data.

(1) `conprep1` for *nodal* data. Comment lines are permitted, where the comment character appears in the first position in the file, (typically, '#'). Four lines of data must appear in the following order:

1. `TYPE, SDSFLAG, EORN` (integer)

   `TYPE = 0`: only generating a table of locations, e.g., x, y, z.

   `TYPE = 1`: obtaining data as well as locations.

   `SDSFLAG < 0`: no SDS file.

   `SDSFLAG = 0`: new SDS file.

   `SDSFLAG = 1`: old SDS file with new attributes and labels.

   `SDSFLAG = 2`: add data with existing attributes and labels. (Usually `SDSFLAG <= 0`, no HDF file or a new one.)

   `EORN = 1`: nodal data.

2. `IMIN, IMAX, JMIN, JMAX, KMIN, KMAX` (integer)

   Gives ranges of nodes defining subdomain of dataset output from `postp`. One pair must be equal to define a plane, and two pairs must be equal to define a line. If an SDS file is being generated, the `IMIN, JMIN, KMIN` values are used to annotate the file, providing a reference frame for offsets in future requests for data from the file.

3. `XSCALE, YSCALE, ZSCALE, VSCALE` (real)

   The first three factors modify the values of the scales, and `VSCALE` modifies the values of the data.

4. `NCHOICE, NDIR` (integer)

   `NCHOICE = 1`: print nodal coordinates as (i,j,k).

   `NCHOICE = 2`: print nodal coordinates as (x,y,z).

   `NDIR = 1`: choice of x-direction for data.

   `NDIR = 2`: choice of y-direction for data.

   `NDIR = 3`: choice of z-direction for data.

(2) **conprep1** for *element* data. Comment lines are permitted, where the comment character appears in the first position in the file, (typically, '`#`'). Five lines of data must appear in the following order, followed by `NGAUS` lines of Gauss point coordinates.

1. `TYPE, SDSFLAG, EORN` (integer)

   `TYPE = 0`: only generating a table of locations, e.g., x, y, z.

   `TYPE = 1`: obtaining data as well as locations.

   `SDSFLAG < 0`: no SDS file.

   `SDSFLAG = 0`: new SDS file.

   `SDSFLAG = 1`: old SDS file with new attributes and labels.

   `SDSFLAG = 2`: add data with existing attributes and labels. (Usually `SDSFLAG <= 0`, no HDF file or a new one.)

   `EORN = 2`: element data.

2. `IMIN, IMAX, JMIN, JMAX, KMIN, KMAX` (integer)

   Gives ranges of nodes defining subdomain of dataset output from **postp**. One pair must be equal to define a plane, and two pairs must be equal to define a line. If an SDS file is being generated, the `IMIN, JMIN, KMIN` values are used to annotate the file, providing a reference frame for offsets in future requests for data from the file.

3. `XSCALE, YSCALE, ZSCALE, VSCALE` (real)

   The first three factors modify the values of the scales, and `VSCALE` modifies the values of the data.

4. `GLYR1, GLYR2, GLYR3` (integer)

   Assuming a $2 \times 2 \times 2$ integration scheme, these values indicate the upper or lower Gauss points in the i, j, and k directions respectively. These values are ignored when 3D data is being extracted.

   `GLYRn = 1`: lower layer in direction n.

   `GLYRn = 2`: upper layer in direction n.

5. `NGAUS` (integer)

This is the number of Gauss points in any element (8).

6. `GP_NUM, XI, ETA, ZETA`

   `NGAUS` lines with the Gauss point number and local (element) coordinates.

(3) `conprep2` — second and final header file for both nodal and element data. Comment lines are permitted. This file contains the label information used by the HDF files. The entry of data is governed by the first non-comment line in the data. All text data is limited to a maximum of 256 characters. The first line contains data for the logical array `LABELS(4)`.

If `LABELS(1)` is true, then three lines of text information are expected:

1. data label;
2. data units;
3. data format for output, e.g., G16.5.

The remaining three array entries in `LABELS` refer to the character data for each of the axes or dimensions of the data. `LABELS(2)`, `LABELS(3)`, `LABELS(4)` respectively indicate the presence of three lines each of data for each axis direction (x, y, z). This data is given on three lines for each case:

1. axis (dimension) label;
2. axis units;
3. axis format for output, e.g., G16.5.

The data labels are modified by the program according to information generated by `postp`, yielding information for this output on the type of data, the time step number, and the simulation time.

7.7.3 *Scripts MKSDSPT and MKSDSPTRUN: for SIM3P3D output*

This script is run using the command:

```
mksds iplast|splast conprep1.hdr conprep2.hdr coords var_num SDSfile
```

The script parameters are explained below.

(i) `iplast|splast` are the output files produced by SIM3D3P, slightly modified at the start with a permutation table and option flags. `iplast` contains pressure, temperature, fluid saturation, porosity, bulk density, and permeability modifier data for each time step. `splast` contains principal stress values and the principal stress number closest to the z- and x-axes respectively, and their actual proximities (as cosine of the angle).

(ii) `conprep1.hdr` is the first part of the control data for `conprep`. The type of information depends on whether one is extracting nodal or Gauss point (element) data—this data is always nodal from SIM3D3P.

(iii) `conprep2.hdr` contains data for labelling and initialising the HDF file.

(iv) `coords` is a file of nodal coordinate data in the form `NODE_NO, X, Y, Z`. Explained in the previous section.

(v) `var_num` is the identification number of the variable being extracted. The current selection of variables is shown in Table 7–2.

`var_num`	`EORN=3`	`EORN=4`
1	pressures (p)	major principal stresses (σ_1)
2	temperatures (T)	intermediate principal stresses (σ_2)
3	oil saturations (S_o)	minor principal stresses (σ_3)
4	water saturations (S_w)	(40) principal stress no. aligned closest to z-axis
		(41) actual proximity as cosine of angle to z-axis
5	gas saturations (S_g)	(50) principal stress no. aligned closest to x-axis
		(51) actual proximity as cosine of angle to x-axis
6	bulk densities (ρ_{bulk})	
7	porosities (ϕ)	
8	permeability ratios ($\frac{k^*}{k}$)	

Table 7–2

Selecting data from the IPLAST and SPLAST files

(vi) `SDSfile` the name of the SDS file to be created and written to.

If a constant IPLAST or SPLAST output file is used, and the `conprep` header files have been set up, a simpler script can be used:

```
mksdsptrun var_num var_name
```

where `var_num` is the identification number of the variable, as above, and `var_name` is the preferred name for that variable (e.g., pressure, prs, dP, dP_cum), and is used as a subscript on the output files. This script simply invokes the other script, `mksdspt`.

7.7.4 *Preparing the* IPLAST *file for conprep*

The IPLAST and SPLAST files generated by SIM3D3P must be slightly modified. The data must appear in the following order:

1. `UCONV, DCUM, DRSEP` (integer)

   `UCONV > 0`: perform default units conversion on data.

   `DCUM = 0`: return absolute values.

   `DCUM = 1`: return changes over the last time step.

   `DCUM > 1`: return cumulative change since the first time step.

   `DRSEP = 0`: return principal stress numbers only.

   `DRSEP > 0`: return cosines of angles only.

2. `NX, NY, NZ` (integer)

   `NX, NY, NZ` are the ranges in the x-, y-, and z-directions respectively. These are output by SIM3D3P automatically.

3. `K, PERMUT(I)` (integer)

   `K`: node number of the permutation vector entry, $1 \leq I \leq (n_x * n_y * n_z)$.

   `PERMUT`: permutation vector relating the fluid and heat flow subdomain to the solid model for which the coordinate data was generated. The permutation vector is available in the output from `grids`, the grid generation program. This line is repeated $n_x * n_y * n_z$ times.

4. `IISVAR(1), IISVAR(2), IISVAR(3), IISVAR(4), IISVAR(5),` (integer)

   `IISVAR > 0`: indicates the inclusion of this variable in the following data set, using the identification scheme outlined in the earlier subsection. This is automatically produced by SIM3D3P.

5. `TIM` (real)

   This is the simulation time for the following data sets, which constitute the bulk of the IPLAST or SPLAST file.

6. `DATA` (real)

   Each data set is written as a vector from SIM3D3P using the ordering $(((p(i,j,k), i = 1, nx), j = 1, ny), k = 1, nz)$. Any node number can be then found from an (i,j,k) reference through the expression $(k-1) * ny * nx + (j-1) * nx + i$. Note that the principal stress direction data is written as $(\sigma_i, \cos\alpha)$, where i is the principal stress number with its direction closest to the z-axis (`varnum = 4`) or x-axis (`varnum = 5`), and α is the angle it makes with the respective axis.

7.7.5 *Preparing the header files for conprep*

(1) `conprep1` header file. As all the variables for SIM3D3P are found at the nodes corresponding to the block centres or point centres, depending on the type of finite difference grid, only one form of the `conprep1` header file is necessary. Comment lines are permitted, where the comment character appears in the first position in the file, (typically, '`#`'). Four lines of data must appear in the following order:

1. `TYPE, SDSFLAG, EORN` (integer)

   `TYPE = 0`: only generating a table of locations, e.g., x, y, z.

   `TYPE = 1`: obtaining data as well as locations.

   `SDSFLAG < 0`: no SDS file.

   `SDSFLAG = 0`: new SDS file.

   `SDSFLAG = 1`: old SDS file with new attributes and labels.

   `SDSFLAG = 2`: add data with existing attributes and labels. (Usually `SDSFLAG <= 0`, no HDF file or a new one.)

   `EORN = 3`: IPLAST data.

   `EORN = 4`: SPLAST data.

2. `IMIN, IMAX, JMIN, JMAX, KMIN, KMAX` (integer)

   These parameters indicate the ranges of nodes defining subdomain of dataset output from `postp`. One pair must be equal to define a plane, and two pairs must be equal to define a line. If an SDS file is being generated, the `IMIN, JMIN, KMIN` values are used to annotate the file, providing a reference frame for offsets in future requests for data from the file.

3. `XSCALE, YSCALE, ZSCALE, VSCALE` (real)

   The first three factors modify the values of the scales, and `VSCALE` modifies the values of the data.

4. `NCHOICE, NDIR` (integer)

   `NCHOICE = 1`: print nodal coordinates as (i,j,k).

   `NCHOICE = 2`: print nodal coordinates as (x,y,z).

   `NDIR = 1`: choice of x-direction for data.

   `NDIR = 2`: choice of y-direction for data.

   `NDIR = 3`: choice of z-direction for data.

(2) `conprep2` — second and final header file for all data. Comment lines are permitted. This file contains the label information used by the HDF files. The entry of data is governed by the first non-comment line in the data. All text data is limited to a maximum of 256 characters. The

first line contains data for the logical array `LABELS(4)`.

If `LABELS(1)` is true, then three lines of text information are expected:

1. data label;
2. data units;
3. data format for output, e.g., G16.5.

The remaining three array entries in **LABELS** refer to the character data for each of the axes or dimensions of the data. `LABELS(2)`, `LABELS(3)`, `LABELS(4)` respectively indicate the presence of three lines each of data for each axis direction (x, y, z). This data is given on three lines for each case:

1. axis (dimension) label;
2. axis units;
3. axis format for output, e.g., G16.5.

The data labels are modified by the program according to information generated by `postp`, yielding information for this output on the type of data, the time step number, and the simulation time.

7.8 *conslice*

`conslice` is a FORTRAN77 program to extract information from a 3D SDS file (generated by `conprep`) generating new SDS files or a table. It is normally used to obtain values lying in a plane and saving them as an SDS (HDF) file. `conslice` does this for every scientific data group (SDG) (i.e., every time step) in the 3D file, saving the data from each SDG to a separate datafile `output.nnn`, where `nnn` is the time step number including leading zeros.

The program consists of the files and subroutines listed in Table 7–3. The program can be easily compiled on a Unix system by typing `make prep` in the presence of the source files and the supplied `makefile`. The program is run using a simple shell script, invoked as follows:

```
mkslice data_file 3D_SDS_source 2D_SDS_output
```

The script parameters are explained in detail below. A higher level script invoking `mkslice` also exists, which will extract n datasets for n different time steps.

```
mknslice type data.hdr data.tail 3D_SDS_prefix 2D_SDS_output tstep(s)
```

The files `data.hdr` and `data.tail` are simply the first and second halves, respectively, of `data_file` used with `mkslice`. The data file is split at the **DTSTEP** entry, which is omitted. The `type` is the suffix used to identify the input 3D SDS file, e.g., **EPS**. `tstep(s)` is a list of time step numbers at

File	*Subroutine*
conslice.f	conslice
ioutils.f	getcmmt echocmmt getstring
sdslice.f	findMaxMin2 sdslice writeslice
setslice.f	setgeom

Table 7–3

Source files and routines constituting CONSLICE

which tables are required.

(i) **`data_file`** is the controlling data used to select a given slice of data in the SDS source file. This information uses the same coordinate system originally used to generate the 3D SDS file. (Offset coordinate values are stored in the 3D SDS file to indicate the coordinate origin when data from a subvolume are saved into the file.) The **`data_file`** format is very simple.

(ii) **`3D_SDS_source`** is the name of the SDS file containing one 3D data group for each time step, complete with annotation, labels and scales.

(iii) **`2D_SDS_output`** is the prefix of the names of the 2D SDS files to which the slice data will be written, one file for each time step. Normally, one specifies a directory in which the files can accumulate, e.g., **`tmp/sdsfile.`**, to which the time step number will be appended as a three-digit sequence.

7.8.1 *conslice data file*

This data file is quite simple as all the information for element or nodal data has been already prepared by **`conprep`** and stored in the 3D SDS file. Comment lines are permitted, where the comment character appears in the first position in the file, (typically, '`#`'). Six lines of data must appear in the following order:

1. `JOB, EORN, NCHOICE` (integer)

    `JOB = 1`: writing new SDS files.

    `JOB = 2`: writing out a table of data at the specified time step.

`EORN` = 1:indicates that the data being read is nodal.

`EORN` = 2: indicates that the data being read is element.

`NCHOICE` = 1: print nodal coordinates as (i,j,k).

`NCHOICE` = 2: print nodal coordinates as (x,y,z).

2. `DTSTEP` (integer)

This is the maximum time step number to be considered. This value is used to generate any data table. This number considers that the first time step is number zero, giving the same numbering scheme used in the original simulation.

3. `IMIN, IMAX, JMIN, JMAX, KMIN, KMAX` (integer)

These parameters indicate the ranges of nodes defining subdomain of dataset output from `postp`. One pair must be equal to define a plane, and two pairs must be equal to define a line.

4. `GLYR1, GLYR2, GLYR3` (integer)

Assuming a $2 \times 2 \times 2$ integration scheme, these values indicate the upper or lower Gauss points in the i, j, and k directions, respectively.

`GLYRn` = 1: lower layer in direction n.

`GLYRn` = 2: upper layer in direction n.

5. `XSCALE, YSCALE, ZSCALE, VSCALE` (real)

Factors to modify values of the scales and the data (`VSCALE`).

6. `LIMFLAG`, (integer)`MAXLIM, MINLIM` (real)

`LIMFLAG` = 0: find local maximum and minimum for each time step and store these values with the pertinent time step.

`LIMFLAG` = 1: find global maximum and minimum across all time steps and store these values with each time step.

`LIMFLAG` = 2: find global maximum and minimum across all time steps, compare these values to `MAXLIM` and `MINLIM`, and store with each time step the most extreme maximum and minimum of the four values.

`MAXLIM`: value used as the global maximum unless the actual global maximum for all the time steps is larger.

`MINLIM`: value used as the global minimum unless the actual global minimum for all the time steps is smaller.

7.8.2 *Hints*

The choice of `LIMFLAG` is important. In the subsequent rasterisation process, the raster values are found by scaling the data values within the range of minimum and maximum values stored for that time step. For meaningful comparisons values of **LIMFLAG = 1** or 2 are the most useful.

7.9 *fp2hdf*

The NCSA utility `fp2hdf` is used to convert the individual SDS files, generated for each time step by `conslice`, into raster images, concatenated into one HDF file. The maximum and minimum values stored in each SDS file are used to calculate the raster values. An 8-bit raster scheme is used, giving 256 different values (0–255). Two end values are reserved, giving only 254 values in practice. All data values are therefore scaled as:

$$v_{raster} = \text{int}\frac{v - v_{min}}{v_{max} - v_{min}} \times 254$$

where v_{raster} is the raster value, v is the data value, and v_{max} and v_{min} are the maximum and minimum values respectively of the dataset. Conversely, a raster value can generate the actual values as

$$v = v_{min} + v_{raster} \times \frac{v_{max} - v_{min}}{254}$$

The following command will generate the raster HDF file:

```
fp2hdf 'tmp/sds.files.' -r -i nx ny
```

where `nx` and `ny` are the number of pixels desired in each direction. The `-r` indicates that a raster file is to be generated. The `-i` option causes bilinear interpolation to be invoked to create the number of pixels. 300 is a typical value. It is preferable to maintain the same aspect ratio as for the extracted data. Once the raster file has been generated, the SDS source files can be removed.

7.10 *Image3*

Once the raster image file has been generated and transferred to a colour Macintosh, the `Image3` program can be invoked (preferably on a Macintosh with 4 Mb or more RAM). The file must be transferred as a binary file. The raster image file is opened as a group. The groups then appear in a scrolling list. A selection of the groups can be made and then opened as a group from the panel displaying the scrolling list. If there is insufficient memory it will have to be done with the `from disk` option. The animation can now proceed. Different palettes offer different perspectives on the data being animated. Absolute values are better shown using a unique range of colours, e.g., the `rainbow` palette. Relative values, i.e., changes, are better illustrated using the complex palettes such as `fire`.

7.11 *Summary*

1. Process the main PLAST-PT3 output file using `postp`.
2. Use `conprep` to extract all data prepared by `postp` to form a 3D SDS (HDF) file.
3. Process the 3D HDF file as desired, using `conslice` to create a 2D SDS file for each time step.
4. Generate a raster image file from all the 2D SDS files using `fp2hdf`.
5. Transfer the raster image file to the Macintosh using a binary transfer, open as a group in `Image3`, and animate.

Chapter VIII

APPLICATION OF THE MODELS

The coupled finite-difference finite-element models have been tested using several of the formulations discussed in Chapter VI. The combinations chosen were:

(i) Three runs based on no feedback to fluid model, studying the effects of

  a) grid refinement in the fluid model; and

  b) injected steam slug size.

(ii) One run based on the staggered formulation.

(iii) One run based on the fully coupled formulation.

The resources required to make each run are considerable. Timesharing constraints meant that the average run would take from two to three weeks to complete, even on the University of Alberta's Convex C210 supercomputer, while several additional days or weeks were required for short runs to determine appropriate control parameters (convergence criteria, iteration limits, compressibility multipliers). On one of the latest RISC workstations offered by the major computer manufacturers, with sufficient core memory, a run should take no more than two or three days — on the assumption that the computer is dedicated to this task alone. This aspect is discussed further at the end of the chapter.

The no-feedback runs are sufficient to gauge the effect of changing the operational parameters (e.g., steam slug size) and the degree of grid refinement in the thermal model on the material response. The fine grid case with the lower steam slug size (lower injection rate for the fixed injection period) was then chosen for comparison using the two different coupled formulations: staggered and fully coupled.

8.1 *Verification of the solid model*

The implementation of the solid model was verified using an example problem from Smith and Griffiths (1982) p. 236. In this example, a cube of material held in an isotropic (confining) stress of −20 kPa is subjected to a prescribed downward displacement over its top face. The cube is fixed with respect to to vertical movements on its bottom face, and two adjoining side faces are fixed with respect to the directions normal to them. These constraints allow the material to extrude from two

Vertical displacements applied to top faces such that plastic deformation occurs. Faces defined by –x, –y, and –z directions not permitted to move in the respective direction.

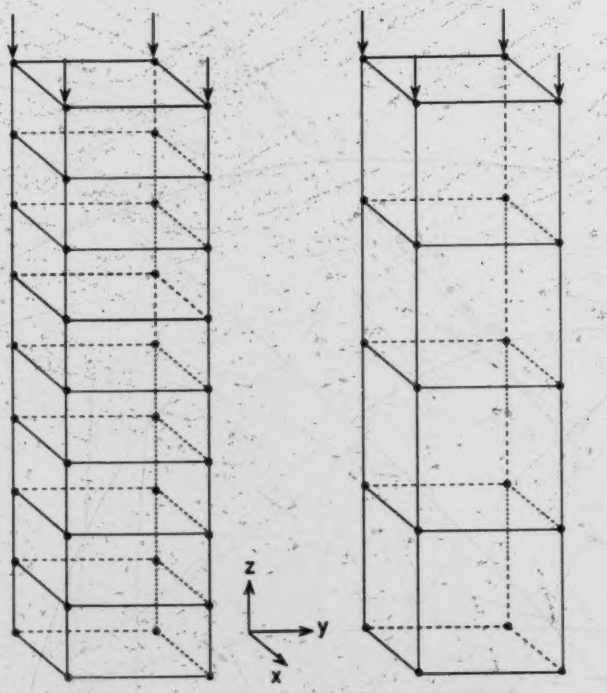


Figure 8–1

1 × 1 × 8 and 1 × 1 × 4 grids used to investigate the effect of grid refinement on the linear brick element, held in an isotropic confining stress field.

unrestrained and adjoining faces, once plastic yielding occurs. The cube is represented by a single linear finite element (8-noded brick). The test was duplicated using two more grids, a $1 \times 1 \times 4$ and a $1 \times 1 \times 8$, as shown in Figure 8–1. Although the new meshes no longer form cubes, the problem remains the same due to the axial nature of the loading. The tests were made for a Mohr-Coulomb material with angle of internal friction $\phi = 30°$ and cohesion $c = 0$, i.e., cohesionless, and elastic properties $E = 2$ MPa, and $\nu = 0.3$. As the load is applied axially, the principal stresses in the unrestrained directions remain constant and equal to the isotropic confining stress, −20 kPa. The axial stress, σ_z, builds up to the maximum permitted for a Mohr-Coulomb material under these

Water compressibility	4.35×10^{-07} kPa^{-1}
Coefficient of thermal expansion of water	0.882×10^{-03} K^{-1}
Oil compressibility	7.25×10^{-07} kPa^{-1}
Coefficient of thermal expansion of oil	0.90×10^{-03} K^{-1}
Rock compressibility	4.35×10^{-07} kPa^{-1}
Specific heat of oil	942.0 kJ/kmol-K
Thermal conductivity of the formation	1.757 kW/m-K
Thermal conductivity of adjacent formations	1.466 kW/m-K
Matrix initial density	2643 kg/m^3
Rock matrix specific heat	0.837 kJ/kmol-K
Thermal diffusivity of adjacent formations	1.0323 m^2/s

Table 8–1

General properties of formation fluids and rock matrix

conditions, -60 kPa (i.e, confining), and remains constant at this value under further axial loading. The same limiting stresses were found for each mesh, while some difference was noted in the values of the displacements and the extent of the plastic zones (as indicated by a positive effective plastic strain). The difference in extent of plastic yield is explained by the greater density of Gauss points at which it is found, which will better resolve the limits of yielding in the body. The distribution of displacements in the body reflects this difference in resolution of plastic strain.

8.2 *Choice of material properties and stress distribution*

A selection of material properties has been made to ensure that the model can be tested without the problem being so difficult as to obscure the interpretation of the results. Accordingly, the model reservoir is given the typical fluid properties of a heavy oil reservoir in eastern Alberta and western Saskatchewan. As a result, the oil is actually mobile at the initial conditions, although that mobility is restricted. The lithological sequence is, however, taken from the Athabasca region of Alberta, and the solid material properties are taken from published data on this region. The regional stresses have been implemented according to a new model which incorporates published field data but permits a varying horizontal stress gradient to be generated from only a few points. This is necessary to account for horizontal stress changes near the surface.

The same properties and conditions have been used for all the runs to permit comparison.

The only variation has been in the steam slug size (i.e., in injection rate) and grid size, and in the coupled formulation.

8.2.1 *Fluid properties*

The general volumetric, thermal, and heat transport properties of the reservoir fluids and the rock matrix are given in Table 8–1.

Initial density of oil	2.243 kmol/m^3
Initial water density	5.590 kmol/m^3
Molar mass of oil	450.0 kg/kmol
Formation temperature	23.3 °C
Separator pressure	101 kPa
Separator temperature	26.7 °C

Table 8–2

General properties of produced fluids

T,° C	S_{wc}	S_{gc}	S_{orw}	S_{org}	S_{gr}	k_{wro}	k_{ocw}	k_{gro}	P_{cw}, kPa	P_{cg}, kPa
23.3	.2000	.0000	.2110	.0900	.0000	.1636	1.0000	.1700	6.895	6.895

Table 8–3

Data for temperature dependent endpoints

The properties of the produced fluids are given in Table 8–2. The relative permeability and capillary pressure data is given in Tables 8–3 and 8–4. Note that the imbibition curve is provided for the oil-water capillary pressure. The general rock and fluid properties at the start of each simulation are given in Table 8–5. There was a small amount of mobile water ($S_{wi} = 0.23$) at the start of each simulation.

8.2.2 *Material properties*

The material properties are listed in Table 8–6. The oil sand zone — the only zone in which fluid pressures and temperature changes were applied, apart from neighbouring elements — was

Water Data				Oil Data			
S_w	k_{rw}	k_{row}	P_{cw}, kPa	S_L	k_{rg}	k_{rog}	P_{cg}, kPa
.20000	.00000	1.0000	.0000	.29000	.17000	.00000	10.342
.25000	.01020	.76900	.0000	.39500	.11200	.02940	7.281
.29400	.01680	.72410	.0000	.43300	.10220	.04610	6.178
.35700	.02750	.62060	.0000	.51500	.08550	.08830	3.785
.41400	.04240	.50400	.0000	.56900	.07610	.11720	2.213
.49000	.06650	.37140	.0000	.61400	.06540	.14330	0.903
.55700	.09700	.30290	.0000	.66300	.05000	.17640	−.524
.63000	.11480	.15550	.0000	.71900	.03720	.21700	−2.158
.67300	.12590	.09560	.0000	.75000	.02850	.22550	−3.0[illegible]1
.71900	.13810	.05760	.0000	.80500	.01950	.29190	−4.661
.78900	.16360	.00000	.0000	.85000	.01210	.33730	−5.971
1.00000	.25000	.00000	.0000	.89900	.00260	.51690	−7.398
				1.00000	.00000	1.00000	−10.342

Table 8–4

Relative permeability data

Initial absolute permeability, k	2.9 μm^2
Initial porosity, ϕ	31%
Initial oil saturation, S_{oi}	0.77
Irreducible water saturation, S_{wirr}	0.20
Residual oil saturation to water, S_{orw}	0.211
Residual oil saturation to gas, S_{org}	0.09
Initial pressure	1380 kPa
Initial temperature	23.3°C

Table 8–5

General rock and fluid properties

assigned representative values for the geomechanical properties of Athabasca oil sand. Values were taken from Dusseault and Simmons (1982), Kosar *et al.* (1987), and Agar *et al.* (1987). The only exceptional value is the initial material density, which should be closer to 2000 kg/m^3 for the oil

Property	Underburden	Oil Sand	Overburden
Elastic Modulus E	30 GPa	2.2 GPa	1 GPa
Poisson ratio ν	0.3	0.3	0.3
Friction angle ϕ	28°	30°	25°
Cohesion intercept c	24 MPa	0 Pa	100 kPa
Material density ρ_r	2600 kg/m3	2500 kg/m3	2500 kg/m3
Coefficient of bulk thermal expansion α_r	5×10^{-5} K^{-1}	5×10^{-5} K^{-1}	5×10^{-5} K^{-1}

Table 8–6

Material properties

sand, but was kept at ca. 2500 kg/m^3 for all the zones in order to make the initial stress distribution simpler.

Linear regression of the volume change data vs. temperature reported in Kosar *et al.* (1987) gave a value of bulk thermal expansion $\alpha_r = 4.5 \times 10^{-5}$ K^{-1}. Prats (1982) suggests values of $11 \pm 1 \times 10^{-5}$ K^{-1} for sandstone and $8 \pm 4 \times 10^{-5}$ K^{-1} for limestone. A value of 5×10^{-5} K^{-1} was used for all the rocks. It is critical that this value be more accurate for the oil sand than for the other two formation types because the increase in horizontal loading which contributes to oil sand yielding (shear failure) comes entirely from thermal expansion. Yielding in the other two formations is not going to affect the outcome of the simulations significantly.

All elements below the oil sand horizon were treated as an underlying limestone with the properties reported from physical testing at the University of Alberta (Chalaturnyk, 1990). The cohesion value was obtained from the reported value of the uniaxial compressive strength, σ_c, which can be related to the cohesion by (e.g., Brady and Brown, 1985, p. 107)

$$\sigma_c = \frac{2c\cos\phi}{1-\sin\phi} \tag{8.1}$$

The reported value of σ_c was between 70 and 80 MPa, which, for the higher value, and an angle of internal friction $\phi = 28°$, gives a value of cohesion $c = 24$ MPa. Typical values for σ_c given in Hoek and Brown (1980), p. 141, for intact limestone vary between 47 and 164 MPa, so the value used in this work is reasonable.

The balance of the material, the overburden, was considered to have the material properties used by Dusseault and Simmons (1982), in addition to the thermal expansion coefficient value described earlier in Table 8-6.

8.2.3 *Stress distribution*

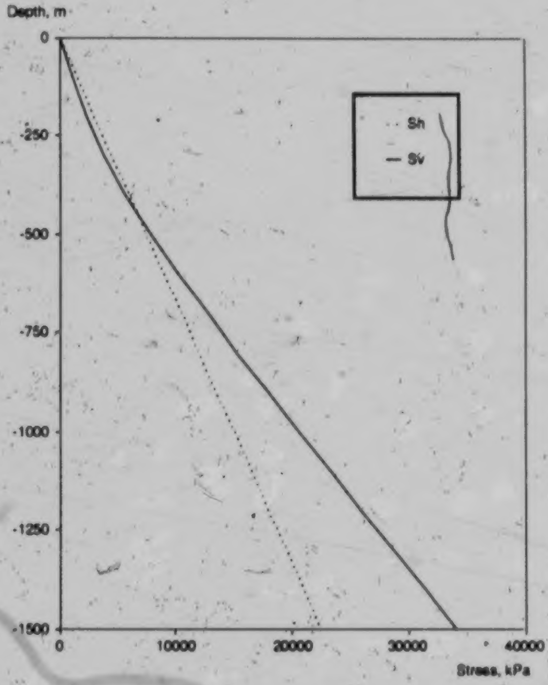


Figure 8-2

Horizontal and vertical stress gradients displaying opposing magnitudes at shallower and greater depths.

Initial stresses in the program are specified in their principal axes using a stress variation with depth chosen to satisfy the local conditions. In most reported cases in hydrocarbon-bearing sedimentary basins, it has been established that the horizontal principal stresses constitute the major and intermediate principal stresses for shallow depths. At deeper levels, the vertical stress is first the intermediate and then the major principal stress. This is why hydraulic fracturing usually will induce horizontal fracture planes at shallow depths, while vertical fracture planes occur at depths below some critical interval. This behaviour is observed, for example, in the Athabasca tar sands

(Dusseault, 1977b). The critical interval extends several hundred metres, in which the major and minor principal stresses are too close together to permit the statement that one or other plane is preferred for a hydraulic fracture to form in. If, however, the shallowest depth at which $\sigma_V = \sigma_1$ is known, then the following simple relationships, developed for this work, can be used:

$$\sigma_{H_1} = a + ez \tag{8.2a}$$

$$\sigma_V = z\left(b - \frac{c}{1 + dz^n}\right) \tag{8.2b}$$

where

(i) z is the depth below surface.

(ii) e represents the average major horizontal stress gradient, assumed constant with depth.

(iii) b is the vertical stress gradient with depth, found for deep levels at a constant value which is greater than the horizontal stress gradient.

(iv) c is an excess gradient parameter to make $\sigma_V < \sigma_{H_1}$ at shallow depths.

(v) d is a matching parameter to ensure that the model satisfies the cross-over depth, z_e, where $\sigma_{H_1} \approx \sigma_V$. d is calculated by equating Equations (8.2a) and (8.2b) at $z = z_e$. The following expression is found for d:

$$d = \frac{a - z_e(b - e - c)}{z_e^n[z_e(b - e) - a]} \tag{8.3}$$

(vi) n is an acceleration parameter to bring the vertical stress gradient σ_V/z to a (relatively) constant value at depths below the cross-over depth. Higher values of n satisfy this requirement as the cross-over depth approaches the surface.

(vii) a is a constant parameter to permit the application of a hydrostatic stress at the surface. All shear stresses must be zero at the free surface unless the material has a cohesive strength; hence, $a = 0$ for soils.

Typical values for some of these parameters are found in Woodland and Bell (1989). E.g.,

$$\left.\begin{aligned} \frac{\sigma_V}{z} &\approx 25\ \text{kPa/m} \\ \frac{\sigma_{H_2}}{z} &\approx 15\ \text{kPa/m} \\ \frac{\sigma_{H_1}}{\sigma_{H_2}} &\approx 1.3 \text{ to } 1.6 \end{aligned}\right\} \tag{8.4}$$

Furthermore, Dusseault (1977b) suggests $\sigma_V \approx \sigma_{H_2}$ at about 300 m, and $\sigma_V \approx \sigma_{H_1}$ at about 500 m, for the Athabasca area. Approximating this as $z_a \doteq 450$ m, and combining the values from Woodland and Bell (1982) with $a = 0$, $c = 15$, and $n = 2$, gives the relation shown in Figure 8–2.

The cross-over of the stress magnitudes can be explained by the presence of less well consolidated sediments near the surface with a lower bulk density than the more deeply buried sediments. This can be accounted for to some degree by reducing the density of the finite elements near and at the surface, but the initial stress model is required to bring the finite element model initial state in line with the regional values. In fact, unless a non-uniform material density distribution is taken into account, such a distribution can cause difficulties under the initial stress method during the initial material settlement in the finite element model. There is also bound to be some horizontal stress relief near the surface, which will reduce the horizontal stress gradient, although not in a manner which affects fracture orientation at reasonable depths (e.g., greater than 200 m). This has not been taken into account in this model, but the omission will not affect the results below the aforementioned reasonable depth. The relationship is purely parametric, but achieves the aim of giving a realistic variation of stresses with depth.

The model studies presented in this thesis use, for simplicity, only the linear relationship of horizontal stress with depth. It is assumed that the horizontal stress field is isotropic, and that the correct vertical stress distribution is found by the initial equilibration the finite element model. The vertical stress is then initially only a function of the material density. Using the overburden density of 2500 kg/m^3, and an initial stress ratio of 0.7 (e.g., see Dusseault and Simmons, 1982), a horizontal stress gradient of -17 kPa/m is suitable. Using the parameters described in this section, this is rendered as $a = b = c = d = 0$ and $e = -17$. The cross-over point is assumed to be at the surface.

8.3 *Fluid model discretizations and operating conditions*

Two fluid model discretizations and two steam slug sizes were used to investigate how the model would react to varying these parameters, and to look for any new insight the results might yield. It was impractical to look at more parameters to vary, given the long times required to obtain the results. In all cases, a cartesian geometry was used which defined one-quarter of the area influenced by a single steam stimulation (injection, soak, and production well). Constant periods of injection, soak, and production were stipulated, so varying the steam injection rate is equivalent to varying the steam slug size.

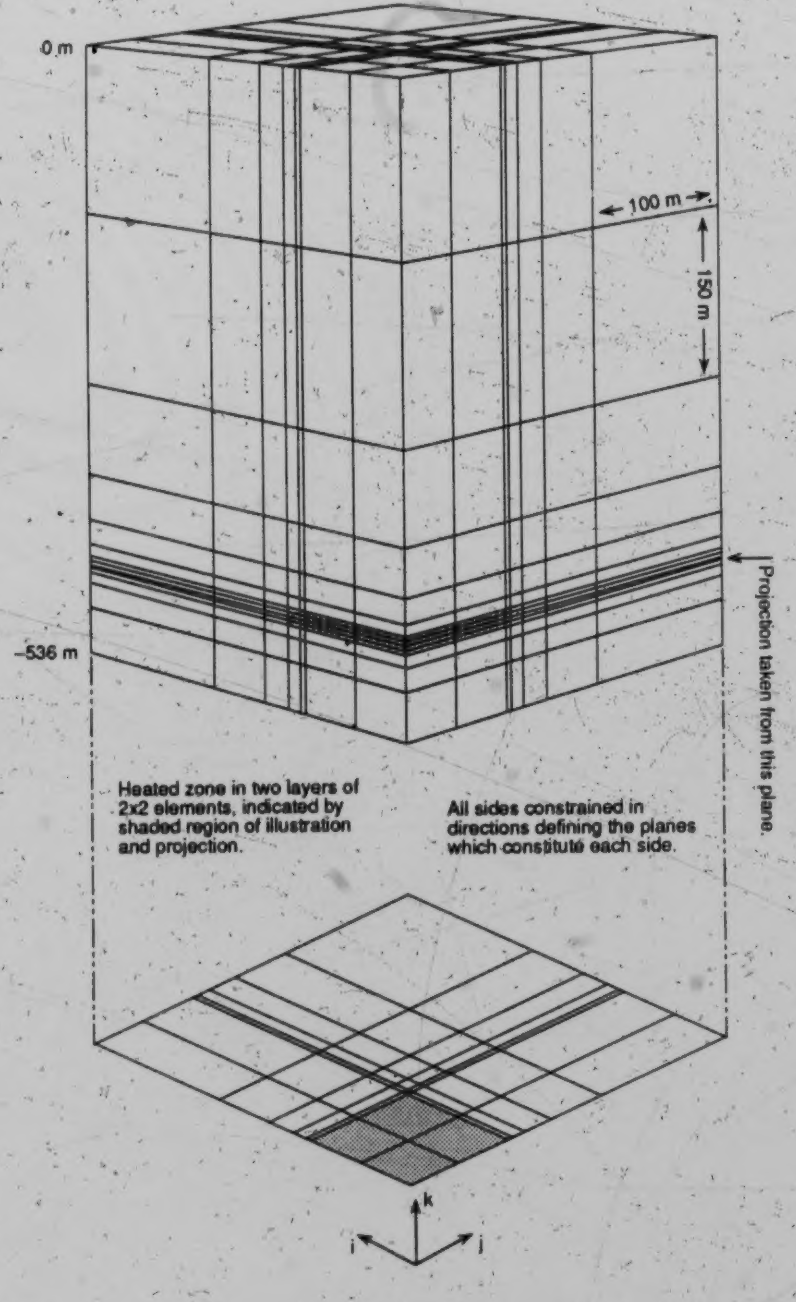

0 m
100 m
150 m
–536 m
Projection taken from this plane.
Heated zone in two layers of 2x2 elements, indicated by shaded region of illustration and projection.
All sides constrained in directions defining the planes which constitute each side.
k
i
j

Figure 8–3

View of simulation domain indicating heated and pressurized zone (3 × 3 × 3 discretization).

The choice of grid for the fluid model — the discretization — influences the choice of grid in the finite element model. A number of elements must surround the heated and pressurized zone, enough to be able to absorb the induced strains and react in a reasonable approximation to an infinite surrounding medium. This is satisfied by strains at the fixed boundary which produce a reaction which is commensurate with the average elastic properties of the undisturbed formations under the existing conditions of stress. Vertically, the surface represents a boundary free to move up and down, so the discretization below the heated and pressurized zone is not so crucial as the lateral surroundings. The discretization in the horizontal plane is generally configured using elements which grow in size by no more than a factor of about two. The final mesh can be generated easily using the `grid` program described in Chapter VII.

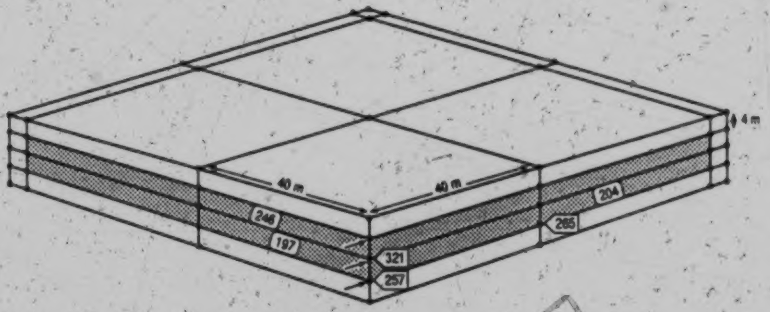


Figure 8–4

View of heated zone and the immediately surrounding elements (3 × 3 × 3 discretization). Injection points with node numbers indicated by arrows; element numbers in lozenges.

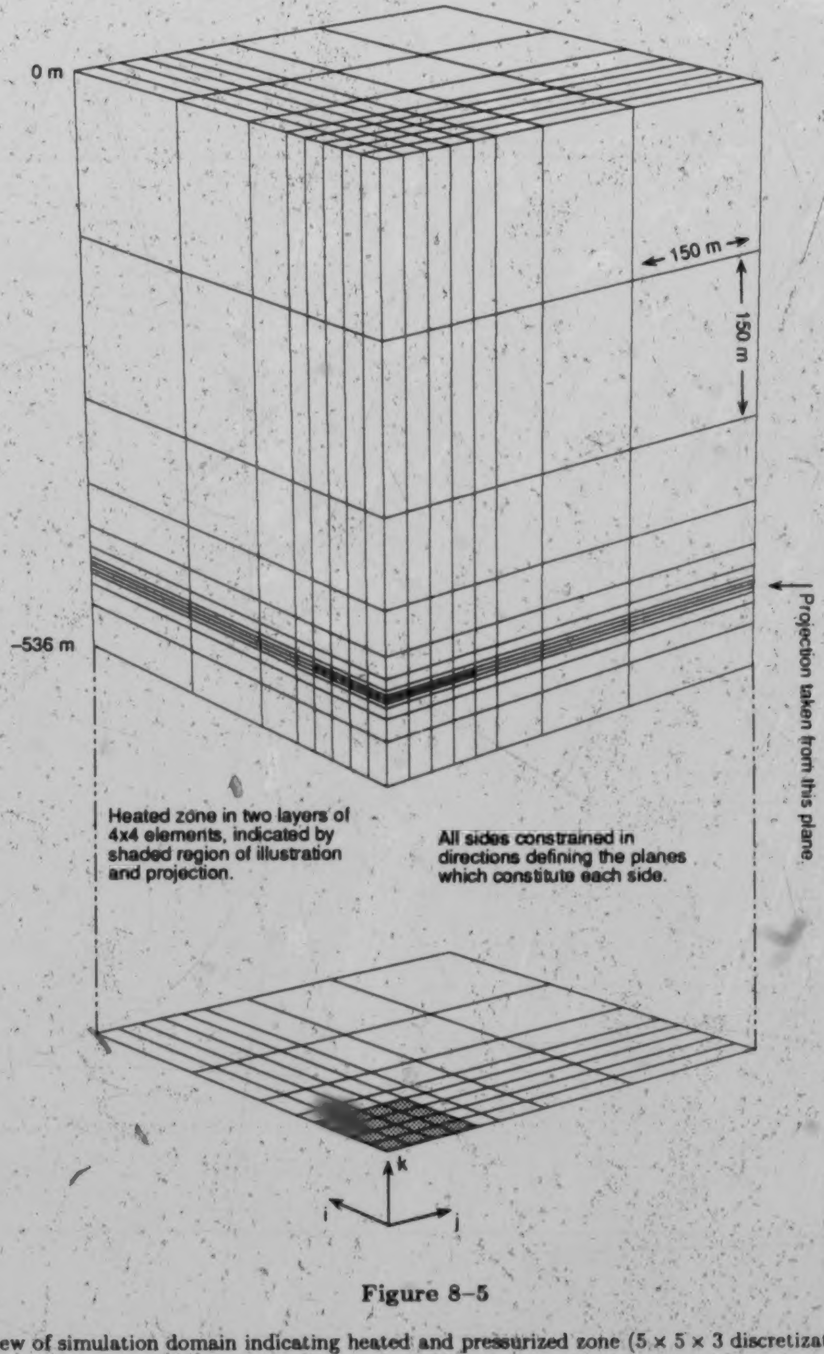
8.3.1 3 × 3 × 3 *fluid model discretization*

This coarse grid was the first used when developing the coupled model. A 3 × 3 × 3 ($x \times y \times z$) finite difference discretization was used to model the domain of heat and fluid flow. Each finite difference block was 40 m by 40 m by 3.7 m; hence, the total grid measured 120 m by 120 m by 11.1 m (393 ft by 393 ft by 36 ft high). This zone, simulated in the finite difference model, is rendered as a 3 × 3 × 3 node system in the finite element mesh, where the nodal values of temperature and pressure corresponded to the block average values given by the finite difference model. This means that the heated region represented by finite elements is a consistent subvolume of the finite domain, with the same aspect ratio, a result of joining a block-centered to a mesh-centered grid. This is not a completely rigorous treatment of the corresponding volumes, but is considered adequate for the purposes of the study. A point-centered finite difference grid would be better choice because it could represent the geometric boundaries exactly on the rectilinear finite element mesh. It would, however, be no better for those blocks not on the boundary of the domain. Properties generated by the finite element results are averaged to the mesh nodes before being used in the finite difference model, ensuring consistency with the finite difference blocks.

The entire heated and pressurized zone was then surrounded by a buffer of elements 4 m (13 ft) thick, in which the applied changes in pressures and temperature varied linearly to zero at the external nodes of this buffer region. This behaviour is a result of the application of the finite element method. The arrangement is illustrated in Figure 8–4, where the shaded region indicates the heated and pressurized zone. The three nodes indicated by arrows correspond to the three injection well block centres used in the finite difference simulation. In the finite element mesh the 3 × 3 × 3 nodes of the heated and pressurized zone effectively become 2 × 2 × 2 elements. These elements were now fitted into the 7 × 7 × 13 element structure shown in Figure 8–3. The finite element grid extends horizontally 254 m in each (x and y) direction, while the vertical extent is clearly marked in Figure 8–3. The two layers containing the heated and pressurized zone were given representative values for the geomechanical properties of Athabasca oil sand. All elements below this horizon were treated as an underlying limestone, and all those above the horizon are treated as the overburden, extending 455 m (1490 ft) to the free surface. It will be noticed that two different geological areas have been juxtaposed to create the simulation domain. The purpose of the simulations was not to emulate some specific region, but rather to illustrate the behaviour of the reservoir sand skeleton and how it might affect the fluid and heat fluxes.

8.3.2 5 × 5 × 3 *fluid model discretization*

This model discretization was chosen to investigate the effect of increasing the mesh density. The incorporation of the 5 × 5 × 3 finite difference blocks into the finite element mesh is as described in the previous subsection. The finite element mesh in this case is 9 × 9 × 14, illustrated in Figure 8–5, where the improvement in refinement over the coarser mesh all lies in the heated and pressurized zone. The block sizes in the finite difference mesh are all equal, 24 m by 24 m by 3.7 m, and the total size remains the same as for the 3 × 3 × 3 example. The finite element buffer zone is laterally larger than for the coarse grid — in this case it is equal to the block size of 24 m. This is illustrated in Figure 8–6. The finite element grid extends horizontally 420 m in each (x and y) direction, while the


0 m
150 m
150 m
–536 m
Projection taken from this plane.
k
i
j

Heated zone in two layers of 4x4 elements, indicated by shaded region of illustration and projection.

All sides constrained in directions defining the planes which constitute each side.

Figure 8–5

View of simulation domain indicating heated and pressurized zone (5 × 5 × 3 discretization).

vertical extent remains the same as in the previous example. The extra lateral extent is an attempt to ensure that the number of elements displaying plastic behaviour remains small as a percentage of the total number of elements.

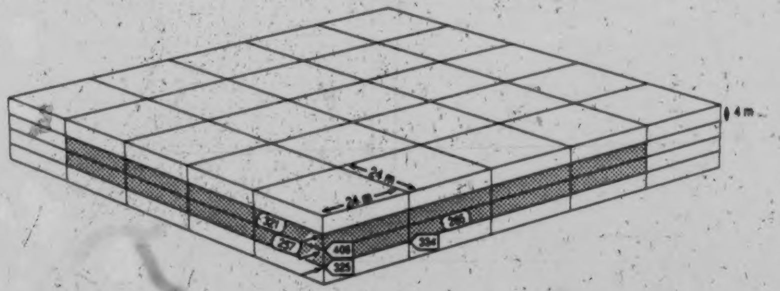


Figure 8–6

View of heated zone and the immediately surrounding elements (5 × 5 × 3 discretization). Injection points with node numbers indicated by arrows; element numbers in lozenges.

8.3.3 *Fluid model operating conditions*

A constant programme of steam injection, soak, and production was considered for each simulation. The schedule is shown in Table 8–7. The higher injection rate was 23.8 m^3/d (150 bpd) of 70% quality steam at 3450 kPa (500 psia). The lower injection rate was 15.9 m^3/d (100 bpd) of steam at the same conditions. The soak period is when the well is shut in, neither injecting nor producing. The production is assumed to come from reducing the bottomhole well pressure to 101 kPa (1 atm), i.e., by pumping the well. A productivity index of 57.65 rm^3/d/MPa (2.5 rb/d/psi) was assigned to the well during this period. The injection rates and production indices must be multiplied by four to get the actual values for the well, as only one quarter of the injection area is under consideration due to symmetry.

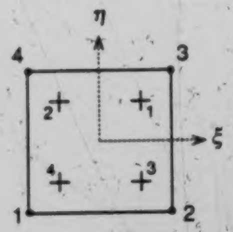
Cycle	Operation	Duration (days)	Cumulative days
1	Inject	30	30
1	Soak	5	35
1	Produce	90	125
2	Inject	30	155
2	Soak	5	160
2	Produce	90	250
3	Inject	30	280
3	Soak	5	285
3	Produce	90	375

Table 8–7

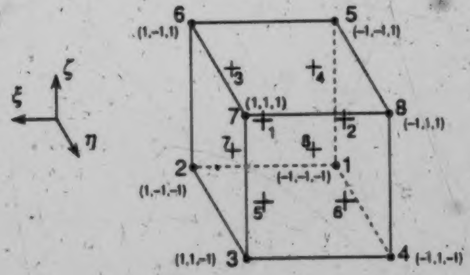
Well operation schedule

8.4 *Results for 3 × 3 × 3 fluid domain with high injection rate*

Results are presented for the high-rate case in the 3 × 3 × 3 grid. Element and node numbers which are referred to are identified in Figure 8–3. Gauss point positions within elements are identified in Figure 8–8. The simulation took 99 time steps to reach 405 days of simulation time. The same number of time steps was used to find the solid model solution. The variation in pressure and temperature with simulation time at the selected nodes 257, 265, and 321 is shown in Figures 8–9 and 8–10, respectively. Nodes 257 and 321 are wellbore nodes, while node 265 is one of the nodes immediately adjacent to the bottom wellbore node, 257.



(a) View of Gauss points in plane.



(b) View of Gauss points in cube element, including numbering relative to the element local node numbering.

Figure 8–8

View of 2 × 2 × 2 Gauss quadrature points inside 8-noded brick elements.

Clearly, big increases in pressure and temperature occur during the three injection cycles with some decline during the soak and production periods. Material behaviour beside these three nodes

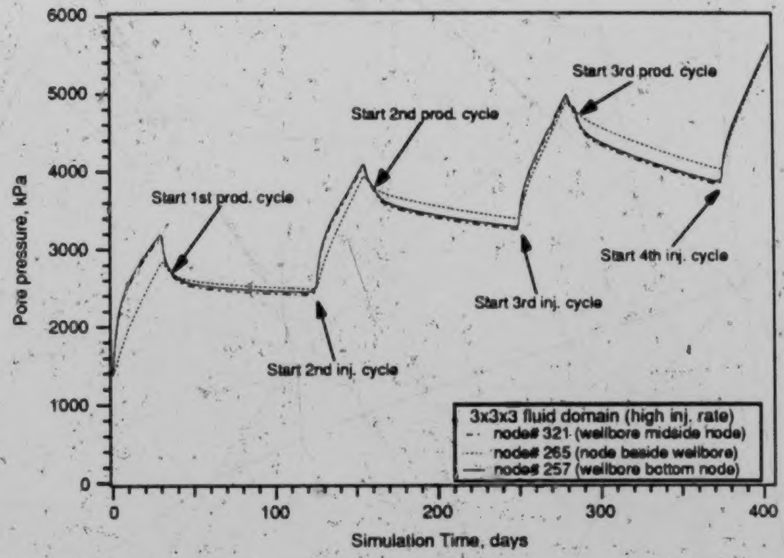


Figure 8–9

Profile of pressure (average block values) with simulation time at selected nodes near wellbore (3 × 3 × 3 fluid domain).

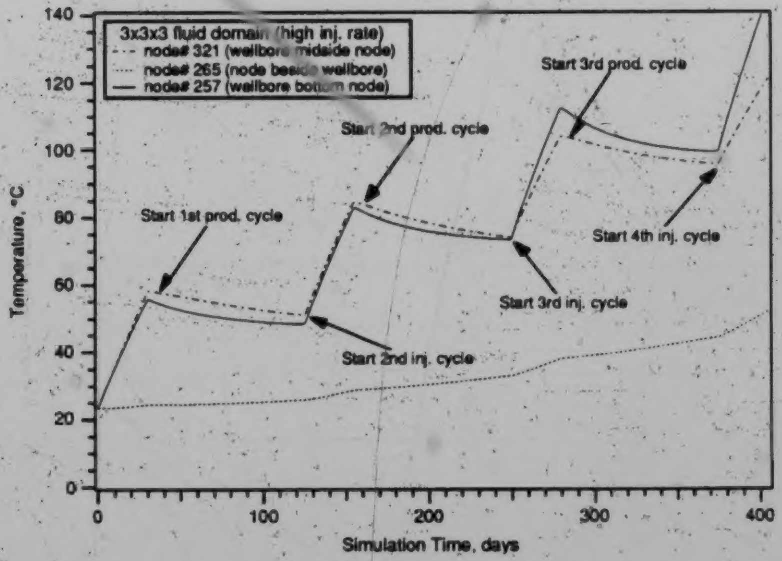


Figure 8–10

Profile of temperature (average block values) with simulation time at selected nodes near wellbore (3 × 3 × 3 fluid domain).

will be presented. Stress-strain behaviour is actually calculated in the finite element model at the Gauss quadrature (integration) points in each element. Three Gauss points from elements 197, 204, and 246 (see Figure 8–3) were chosen such that they were the ones closest to the nodes reported in Figures 8–9 and 8–10. The data points are organized in the next five diagrams so as to correspond to the nodes considered for the pressure and temperature profiles. The nodes and their closest Gauss points will be referred to interchangeably in the following text. Five main quantities are reported: effective plastic strain $d\bar{\epsilon}_p$, volumetric strain ϵ_v, stress ratio K_o, vertical stress σ_{zz}, and Lode angle θ. These quantities have all been defined in a previous chapter, but K_o will be reviewed. Values of minor principal stress and its orientation are also reported, as is the porosity of the formation.

The stress ratio, K_o, reports the ratio of the major horizontal stress to the vertical stress. Note that in the tension-positive stress definition, it is actually the minimum compressive horizontal stress by magnitude (min $|\sigma_{H_1}, \sigma_{H_2}|$). Values less than one indicate that a hydraulic fracture or parting is most likely to be vertical, while values greater than one indicate that a horizontal parting is more likely. The vertical stresses at the selected Gauss points can be combined with the stress ratio to postulate the initiation of hydraulic fractures.

Figure 8–11 illustrates the growth in plastic strain at the three Gauss points. The trend follows the injection cycles closely, but plastic straining does not begin until the simulation is into the second injection cycle. The bottom well node shows the greatest degree of plastic strain. This is because of the higher vertical stress at the deeper point, combined with the high temperature and pressure changes at the injection point. The points which have yielded tend to accumulate more strain than any surrounding points until the stresses have been sufficiently disturbed farther away. Figure 8–12 illustrates the increase in volumetric strain as injection continues. Initially, the volumetric strain increase is due to elastic relief as the pore pressure builds up, with some fall-off during the pressure decreases accompanying the soak and production periods. Thus the values for both injection points (elements 197 and 246) are initially very close, while in the second and third production cycles, a gap opens up between the two points, probably because of the greater amount of plastic strain occurring at the lower node. The final degree of volumetric strain at the end of the simulation is not much over 0.6%, which would not cause much permeability enhancement. However, the plastic straining can be a very local phenomenon, and this low value highlights the coarse nature of the finite element mesh used, spreading a larger amount of local strain over a greater volume.

Figure 8–13 illustrates the change in the stress ratio at each of the selected Gauss points. The values reflect the pressure gradients between the nodes, the distributed loading caused by the change in temperature over the time step, and the degree of accumulated plastic deformation. The nodes

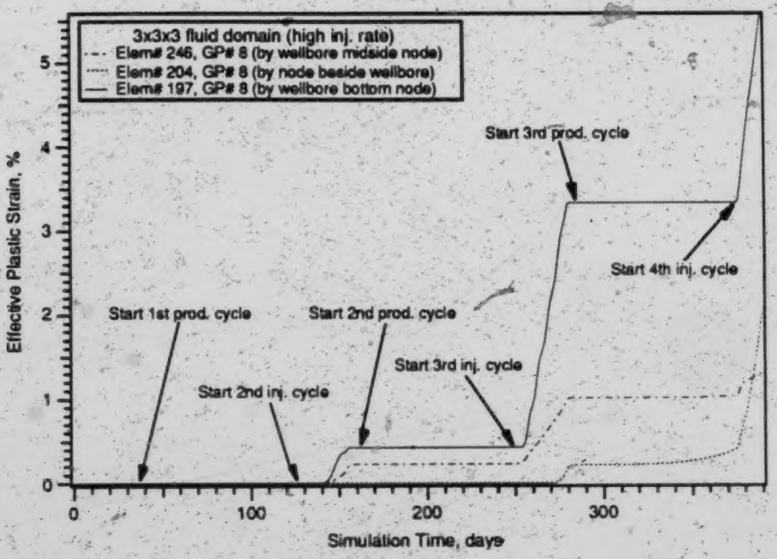


Figure 8–11

Growth of effective plastic strain with simulation time at selected Gauss points near the wellbore (3 × 3 × 3 fluid domain).

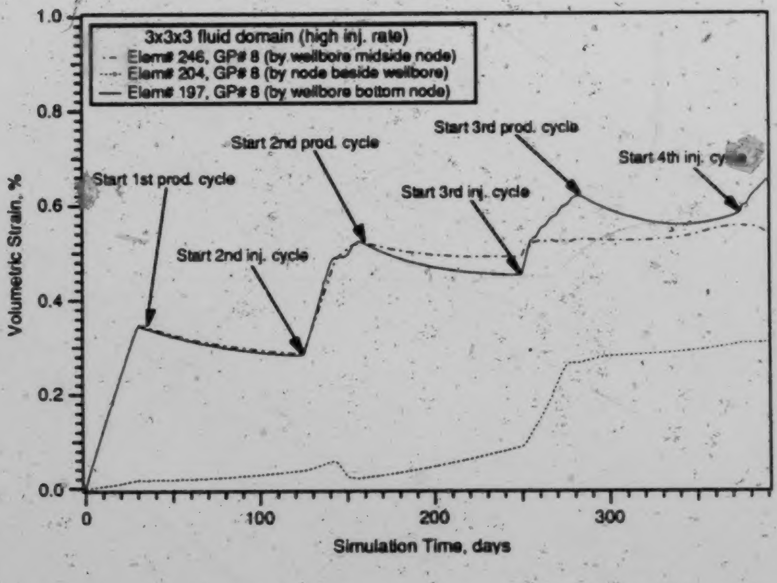


Figure 8–12

Growth of volumetric strain with simulation time at selected Gauss points near the wellbore (3 × 3 × 3 fluid domain).

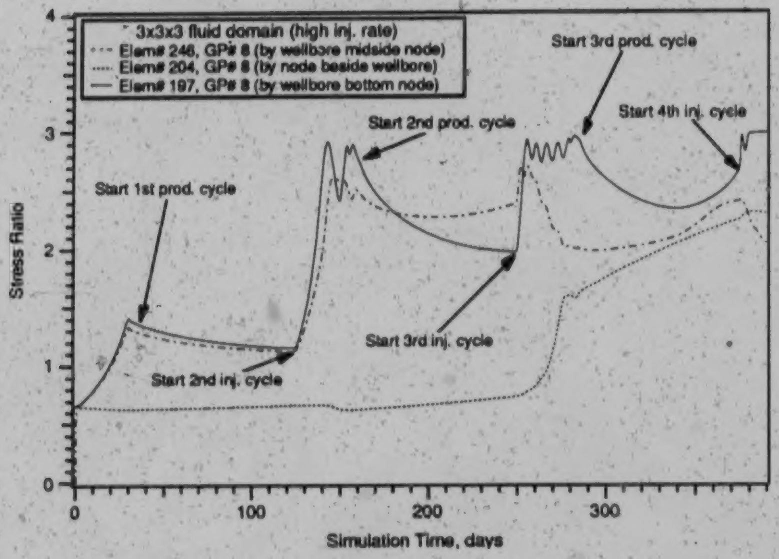


Figure 8–13

Change in stress ratio, K_o, with simulation time at selected Gauss points near the wellbore ($3 \times 3 \times 3$ fluid domain).

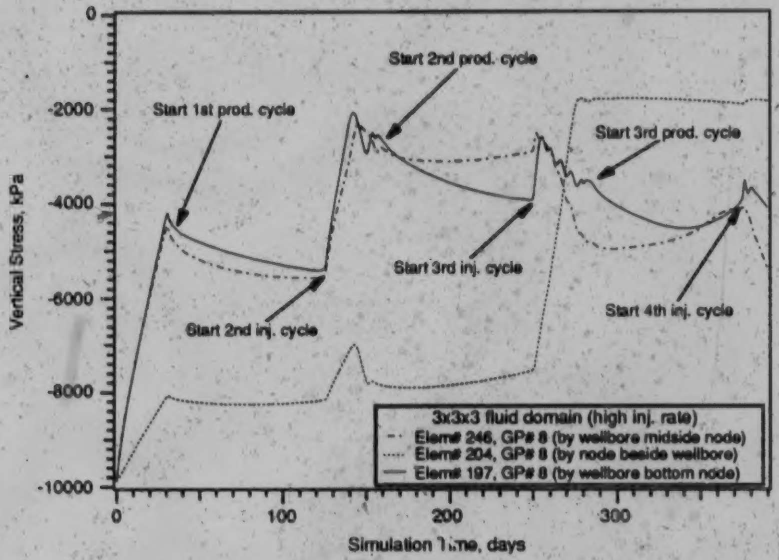


Figure 8–14

Change in vertical stress with simulation time at selected Gauss points near the wellbore ($3 \times 3 \times 3$ fluid domain).

at the wellbore quickly move to values of K_o above one during the first injection cycle, indicating that a horizontal orientation would be preferred in the event of a (tensile) hydraulic fracture. (σ_V must be the minimum principal stress by magnitude if $K_o > 1$.) The condition at the neighbouring Gauss point, in element 204, differs in that the stress ratio does not go above one until the third injection cycle. This corresponds to the increase in volumetric strain at this point which can be seen in Figure 8–12. The degree of disturbance in K_o at the two injection nodes as a result of changing pressure and temperature is clearly illustrated by comparison with the pressure and temperature profiles. Figure 8–14 complements the information given in Figure 8–13. The vertical stresses for the selected Gauss points are shown to vary considerably with the changing pressure and temperature. Compressive stresses are considered to be negative in continuum mechanics, so the values of σ_{zz} must move to zero or positive (tension) values to cause a fracture. The results indicate that this never happens at these points during the simulation. The response at the neighbouring node, indicated by element 204, is particularly interesting. By the end of the third injection cycle the magnitude of the vertical stress at this point appears to have decreased the most. This is mirrored in the plot of stress ratio vs. time at the same point in Figure 8–13. It would appear that this related as much to the neighbouring events of deformation as to the changing values of pressure and temperature. For example, the third injection cycle also corresponds to the significant increase in effective plastic strain at the bottom wellbore node. From the existing trend, it is likely that the point would have moved into tension in a fourth injection cycle (just begun at end of the study), thereby indicating that a horizontal parting had occurred.

The next illustration, Figure 8–15, shows the Lode angle, θ, varying with time. It clearly illustrates the rotation of the principal stresses by moving from +30° to −30° very quickly. The intermediate values indicate that the intermediate principal stress has changed relative to the major and minor principal stresses.

The issue of principal stress rotation causes problems in acquiring relevant information on material behaviour. Classical geomechanical testing of a sand assumes an undisturbed sample at the outset of a triaxial compression test, excluding specific tests for heterogeneous materials. Forcing the material to yield by following a certain stress path induces new heterogeneity into the material as grains slide over each other in preferred directions. Moreover, grain crushing might occur, and localized zones of disturbance appear. It is not reasonable to then suppose that the application of a subsequent loading path, highly deviated from the original, can be modelled using the same material parameters derived from the original loading of the undisturbed sample. This problem has been considered very recently by Pietruszczak and Krucinski (1989), but is beyond the scope of most

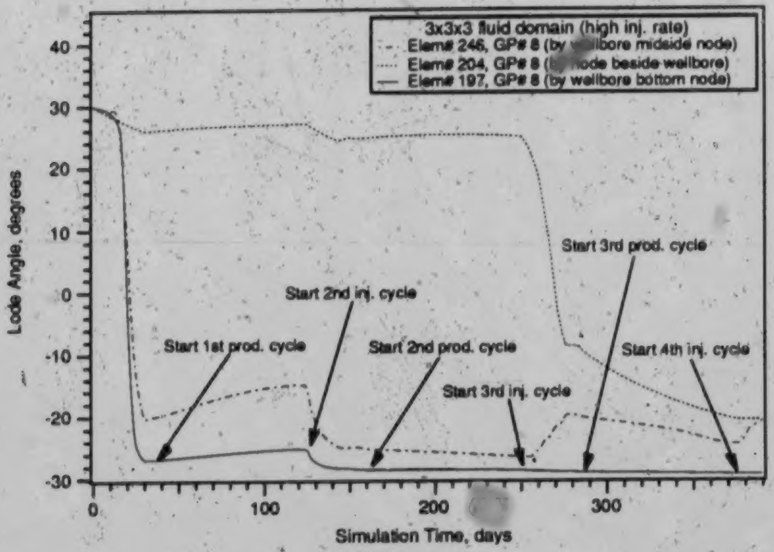


Figure 8–15

Change in Lode angle, θ, with simulation time at selected Gauss points near the wellbore ($3 \times 3 \times 3$ fluid domain).

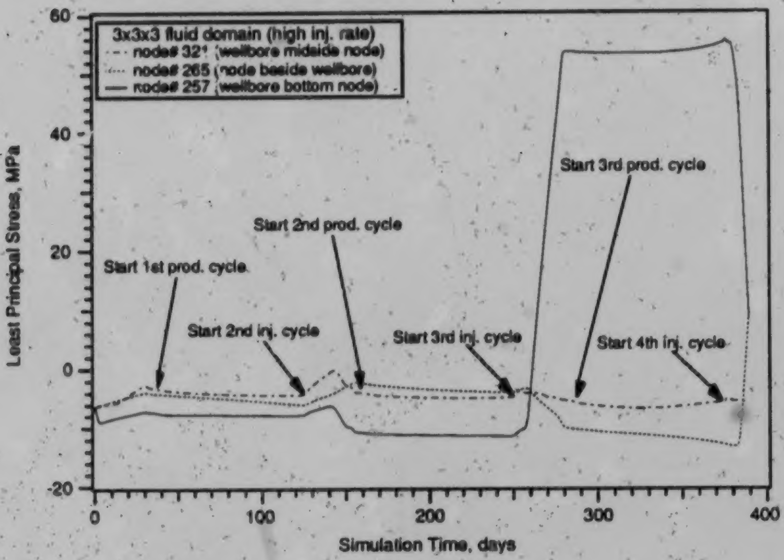


Figure 8–16

Change in least principal stress, σ_3, with simulation time at selected nodes near the wellbore ($3 \times 3 \times 3$ fluid domain).

current work. The treatment of these changes has been neglected in this work, and can therefore be considered a limitation.

On the same page, Figure 8–16 illustrates the variation of minor principal effective stress (in the compression-positive sense) with time at the same selected nodes as the earlier pressure and temperature profiles. This value was obtained from stresses approximated at the nodes using the scheme described in §6.5.2. The results illustrate the problem of extrapolating the solutions to the boundary from only one set of element points, as is the case at the corner nodes where steam injection takes place. The stresses at the Gauss point (as in the vertical stress reported in Figure 8–14) are the most accurate values within the element, insofar as the error is minimized at the quadrature points. The Gauss points can be shown to be the optimum sampling locations for stresses (Zienkiewicz, 1977). The variation of σ_3 at node 257, the bottom injection point, indicates that a tensile parting stress occurs shortly after the start of the third injection cycle. This value is found by extrapolating the variation in stress within the element to the node. Obviously, the material cannot support positive (tensile) stresses and the value is an artifact of the mesh size and the lack of contribution from neighbouring elements. The other two nodes have values which are in accord with the Gauss point results presented for the vertical effective stresses.

Two more figures illustrate the behaviour of the principal stresses at the nodes, found from the eigenvector solutions to the stress characteristic polynomial. Figure 8–17 indicates which principal stress is closest in orientation to the vertical axis. This is of interest because a horizontal parting is known to be the most effective for heating a sand saturated with an immobile oil. All the nodes start out with the major principal stress being aligned with the vertical axis. In the first injection period, both the well injection nodes rotate the principal stresses so that the vertical axis coincides with the minor principal stress. The principal stress rotation occurs later at the neighbouring nodes, occurring in the second injection cycle at node 265. As a parting will orient itself normal to the least principal stress, these results indicate that the conditions for a horizontal hydraulic parting are induced by the injection of steam. The actual proximity of the closest principal stress to the vertical axis is indicated in Figure 8–18, where the cosine of the angle between the closest principal stress and the z-axis is presented. The principal stress and vertical axes coincide at the two injection nodes, while there is some rotation of the principal stresses away from the geometric axes at the neighbouring node until the start of the third injection cycle. The principal cause of the stress rotation is the induced straining from heating the formation.

The variation in porosity near the wellbore is presented in Figure 8–19. The variation is identical to the change in pressure shown in Figure 8–9 because the porosity is a linear function

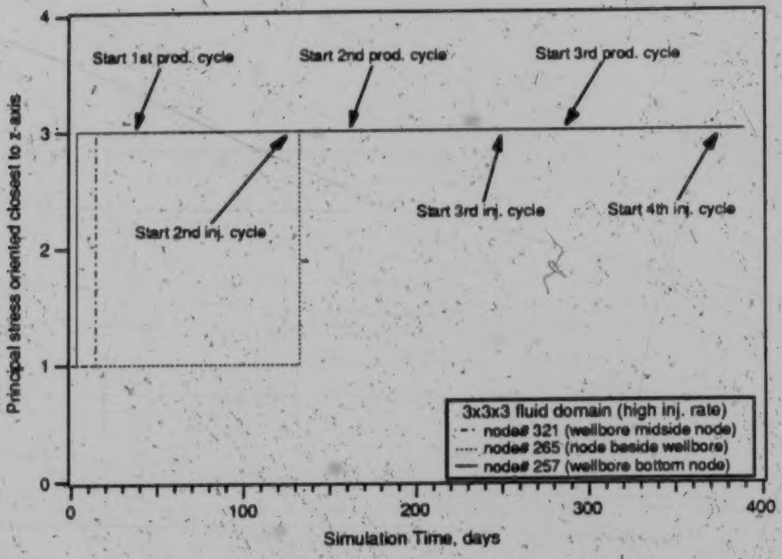


Figure 8–17

Change in the principal stress number oriented closest to the z-axis at selected nodes near the wellbore (3 × 3 × 3 fluid domain).

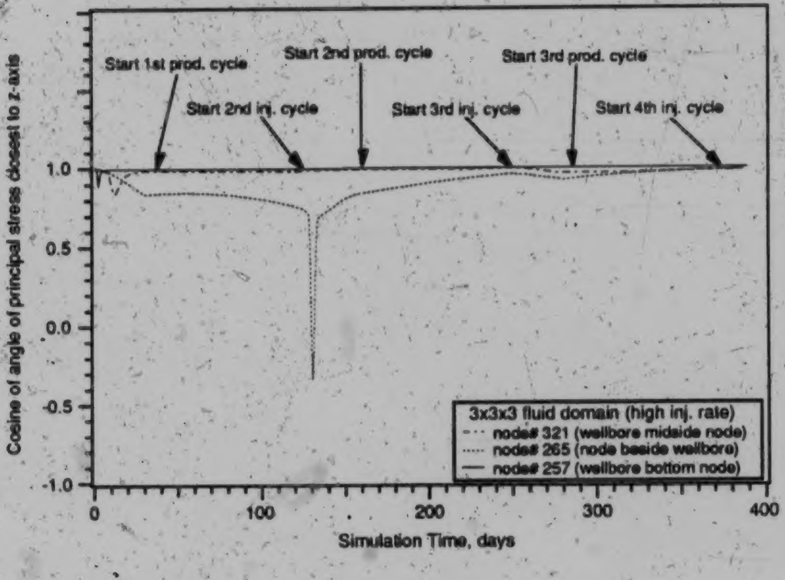


Figure 8–18

Change in the angle of the principal stress oriented closest to the z-axis at selected nodes near the wellbore (3 × 3 × 3 fluid domain).

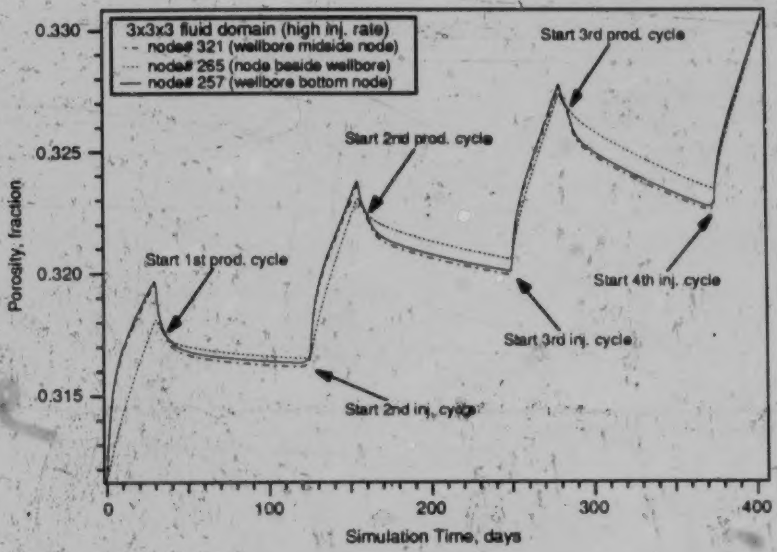


Figure 8–19

Change in porosity, ϕ, with simulation time
at selected nodes near the wellbore ($3 \times 3 \times 3$ fluid domain).

of pressure in the uncoupled model. As such, the illustration is unremarkable except for later comparison with the coupled studies. This is the classic treatment of rock matrix compressibility in petroleum reservoir simulation.

8.5 *Results for* 5 × 5 × 3 *fluid domain with high injection rate*

Results are presented for the high-rate case (uncoupled) in the 5 × 5 × 3 grid. Element and node numbers which are referred to are identified in Figure 8–5. Gauss point positions within elements are identified in Figure 8–7. The simulation took 160 time steps to reach 405 days of simulation time. The same number of time steps was used to find the solid model solution. The variation in pressure and temperature with simulation time at the selected nodes 325, 334, and 406 is shown in Figures 8–19 and 8–20, respectively. Nodes 325 and 406 are wellbore nodes, while node 334 is one of the nodes immediately adjacent to the bottom wellbore node, 325.

Clearly, big increases in pressure and temperature occur during the three injection cycles with some decline during the soak and production periods. As before, material behaviour beside these three nodes will be presented. Three Gauss points from elements 257, 321, and 265 (see Figure 8–5) were chosen such that they were the ones closest to the nodes reported in Figures 8–19 and 8–20. The data points are organized in the next five diagrams so as to correspond to the nodes considered for the pressure and temperature profiles. The nodes and their closest Gauss points will be referred to interchangeably in the following text. The same terms reported for the 3 × 3 × 3 fluid domain study will be presented in this section.

Figure 8–21 illustrates the growth in plastic strain at the three Gauss points. The trend initially follows the injection cycles closely, although plastic straining does not begin until the simulation is into the second injection cycle. The bottom well node shows the greatest degree of plastic strain. The amount of plastic strain becomes excessive after the onset of the third injection cycle. Small strain analysis should be restricted to (extensional) deformations of not much more than 5%, although the error associated with exceeding this suggested limit is only 15% at an engineering strain value of 40% (Annigeri *et al.*, 1985, comparing logarithmic to engineering strain in extensions). The large values reported here arise because the stresses at plastic Gauss points have not been permitted to be reduced back to the yield surface. The zero hardening parameter of the material properties chosen for the oil sand make it necessary to iterate the solid solution many more times than for a material with a positive hardening parameter. (See Appendix F, §F.2.5.) Each iteration is an attempt to reduce the excess trial stress back to the yield surface, which is the confining boundary for stresses in the material, to the so-called contact stress. A maximum of 20 iterations was permitted for each load step in the solid model, where perhaps several thousand were necessary in some instances. The low maximum is necessary so that the computer time required does not become inordinately large. In this case, the solid results are only marginally useful after about half way through the third injection cycle. The stress reduction algorithm used is the simple one reported in Owen and

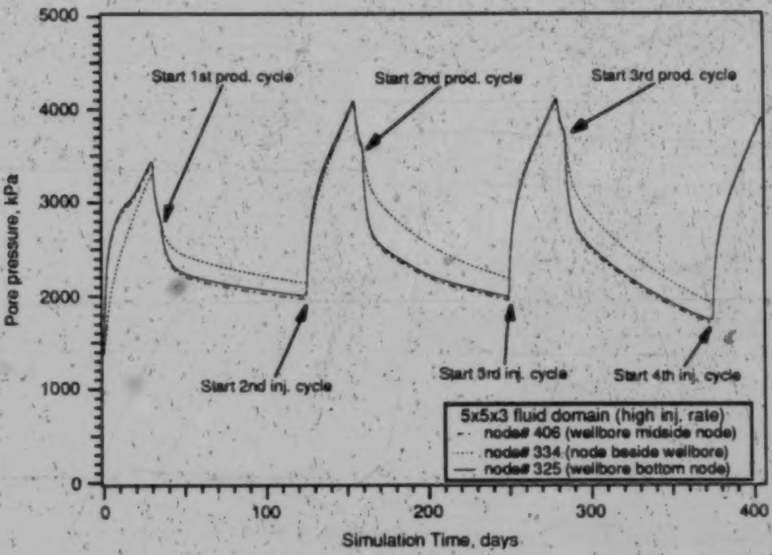


Figure 8–19

Profile of pressure (average block values) with simulation time at selected nodes near wellbore (5 × 5 × 3 fluid domain; high rate).

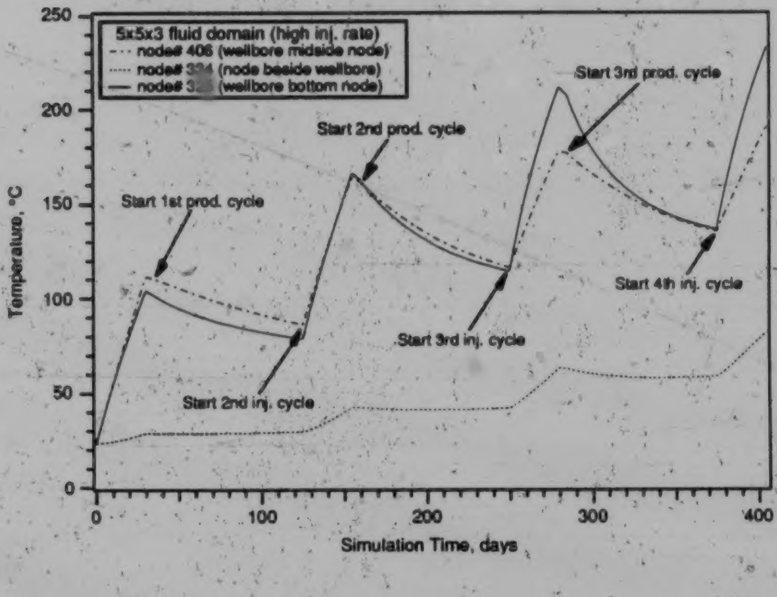


Figure 8–20

Profile of temperature (average block values) with simulation time at selected nodes near wellbore (5 × 5 × 3 fluid domain; high rate).

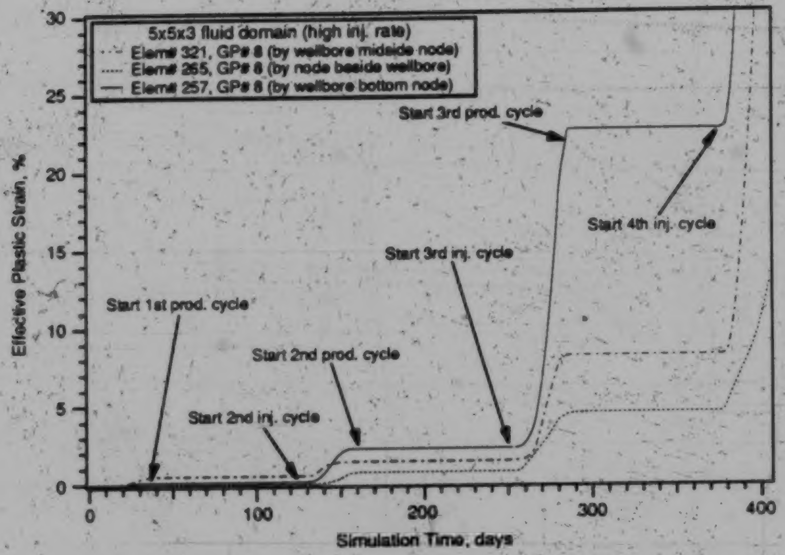


Figure 8–21

Growth of effective plastic strain with simulation time at selected Gauss points near the wellbore (5 × 5 × 3 fluid domain; high rate).

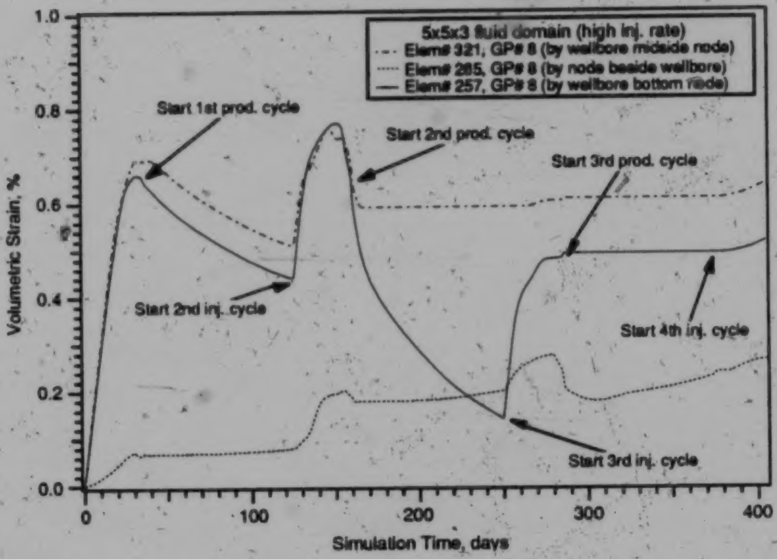


Figure 8–22

Growth of volumetric strain with simulation time at selected Gauss points near the wellbore (5 × 5 × 3 fluid domain; high rate).

Hinton (1982). Better approaches are available to reduce the excess stresses to the yield surface, but the simple stress reduction algorithm is still useful in these simulations, where the objective is to demonstrate the feasibility of making coupled reservoir simulation and geotechnical studies.

Figure 8–22 illustrates the variation in volumetric strain as injection continues. The flat response near the middle injection point (element number 321) after the second production cycle commences is probably due to the excessive plastic strain reported earlier. The most anomalous value is that of the decline of ε_v at the bottom injection point (element 257) during the second production cycle. This is connected with the large (negative) vertical stresses at the same time, which probably also arise due to the incomplete plastic solution at points in the mesh.

Figure 8–23 illustrates the change in the stress ratio at each of the selected Gauss points. As before, the nodes at the wellbore quickly move to values of $K_\circ$ above one during the first injection cycle, indicating that a horizontal orientation would be preferred in the event of a (tensile) hydraulic fracture. The condition at the neighbouring Gauss point, in element 265, differs in that the stress ratio does not go above one until the third injection cycle. The results become doubtful during the second production cycle due to the incomplete plastic solution, where high vertical and horizontal stresses remain unresolved. Nevertheless, this information will be presented because some insight into events still can be extracted. This behaviour of the stress ratio corresponds to the decrease in volumetric strain at the same point which can be seen in Figure 8–22. Figure 8–24 complements the information given in Figure 8–23. The vertical stresses for the selected Gauss points closest to the injection nodes display extreme values at a point half way through the second injection cycle. These values are attributed to the unresolved stress state.

The next illustration, Figure 8–25, shows the Lode angle, θ, varying with time. It clearly illustrates the rotation of the principal stresses during the first injection cycle by moving from +30° to −30° very quickly. The unresolved stress state half way through the second injection period manifests itself as a change in the Lode angle to values close to zero, which represents a state of pure shear.

On the same page, Figure 8–26 illustrates the variation of minor principal effective stress (in the compression-positive sense) with time at the same selected nodes as the earlier pressure and temperature profiles. These values, as before, were obtained from stresses approximated at the nodes using the scheme described in §6.5.2. The problem of extrapolating the solutions to the boundary from only one set of element points is exacerbated by the large stress gradients existing in the elements with unresolved stresses. This shows up in the saw-tooth outline of the approximated minor principal stress at node 325 during the second injection cycle. However, the early time data

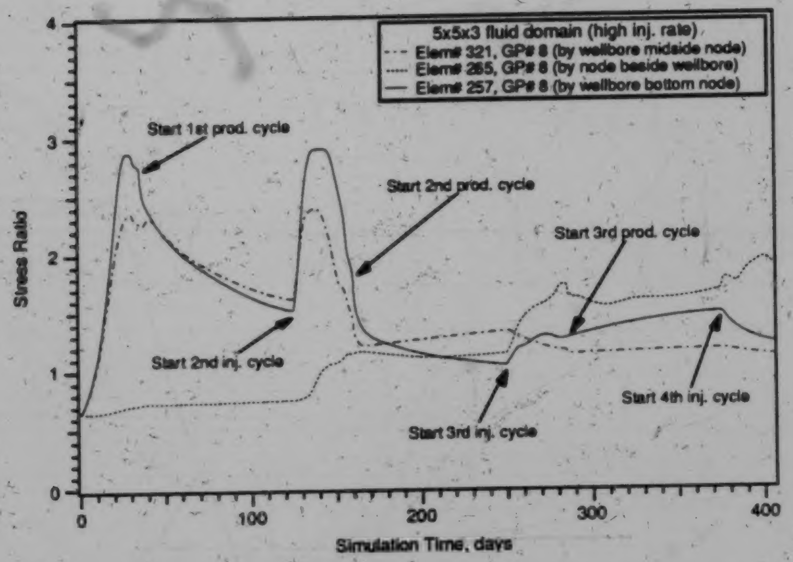


Figure 8–23

Change in stress ratio, K_o, with simulation time at selected Gauss points near the wellbore (5 × 5 × 3 fluid domain; high rate).

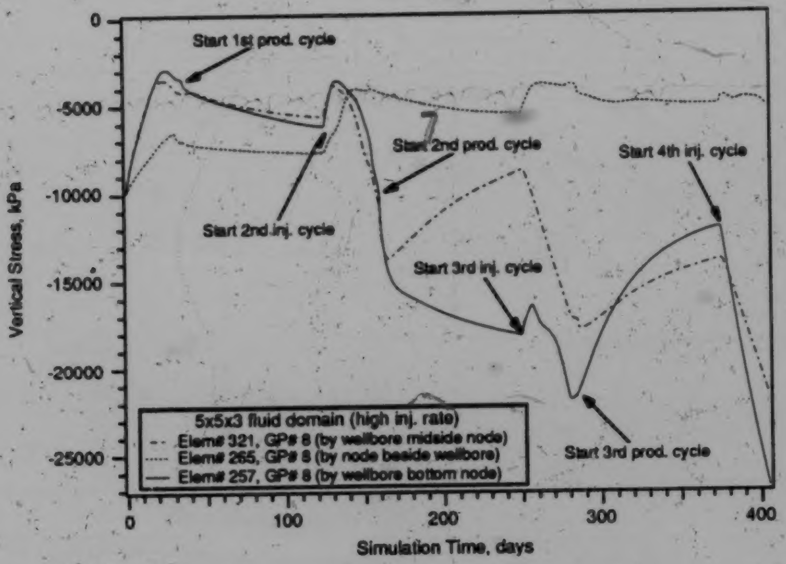


Figure 8–24

Change in vertical stress with simulation time at selected Gauss points near the wellbore (5 × 5 × 3 fluid domain; high rate).

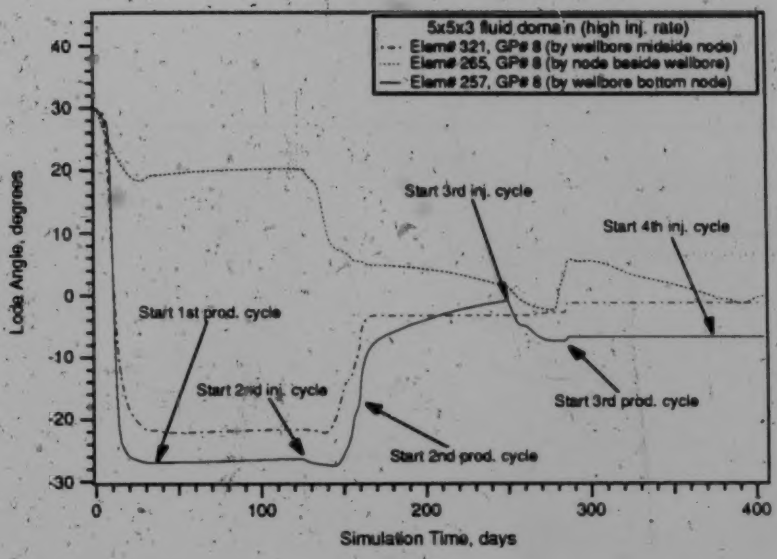


Figure 8–25

Change in Lode angle, θ, with simulation time at selected Gauss points near the wellbore ($5 \times 5 \times 3$ fluid domain; high rate).

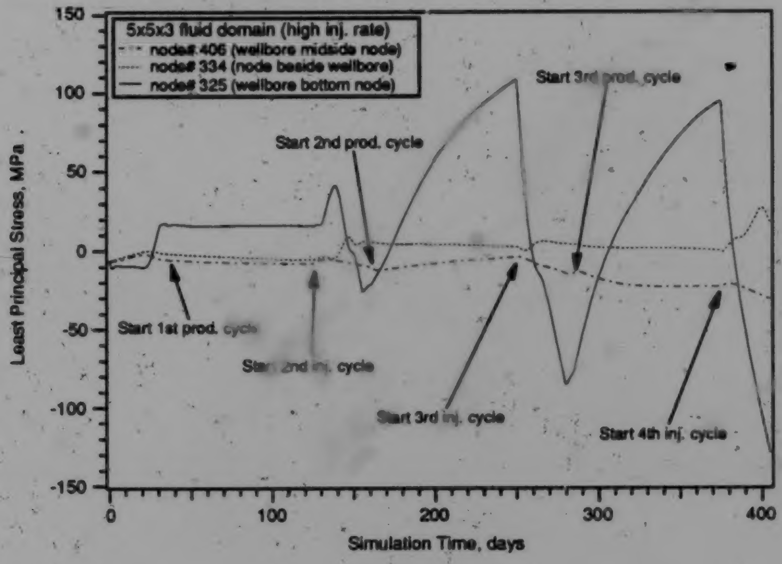


Figure 8–26

Change in least principal stress, σ_3, with simulation time at selected nodes near the wellbore ($5 \times 5 \times 3$ fluid domain; high rate).

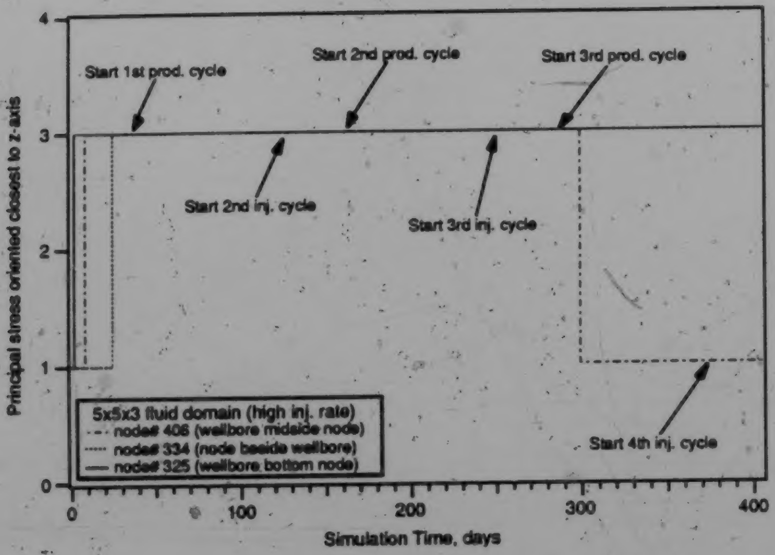


Figure 8–27

Change in the principal stress number oriented closest to the z-axis at selected nodes near the wellbore (5 × 5 × 3 fluid domain; high rate).

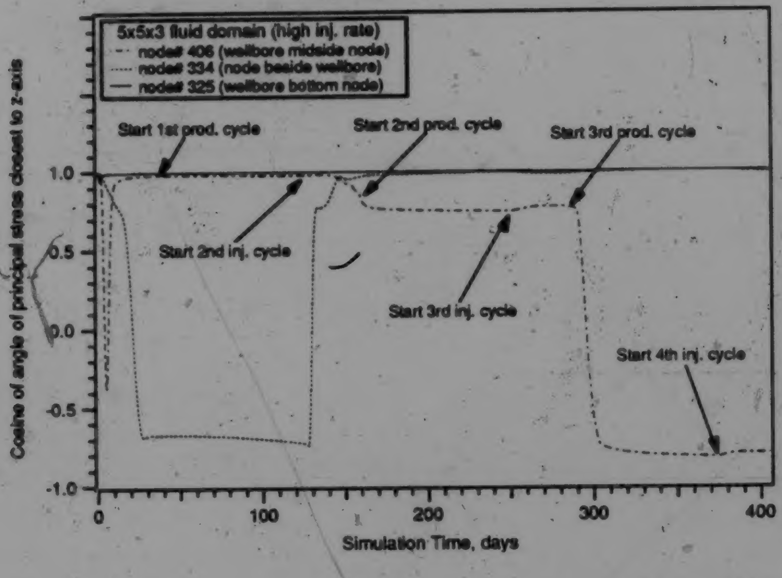


Figure 8–28

Change in the angle of the principal stress oriented closest to the z-axis at selected nodes near the wellbore (5 × 5 × 3 fluid domain; high rate).

is still useful, and it can be seen that σ_3 at the node goes into tension during the first production cycle, indicating that the conditions for a hydraulic fracture have been met. Comparing this with the stress ratio, or the principal stress number in the later diagram, a horizontal parting would occur because the least principal stress coincides with the vertical axis.

Two more figures illustrate the behaviour of the principal stresses at the nodes, found from the eigenvector solutions to the stress characteristic polynomial. Figure 8–27 indicates which principal stress is closest in orientation to the vertical axis. All the nodes start out with the major principal stress being aligned with the vertical axis. In the first injection period, all the nodes under consideration rotate the principal stresses so that the vertical axis coincides with the minor principal stress. The neighbouring node has principal stress rotation in the first injection period, which did not occur in the 3 × 3 × 3 mesh until the second injection period.

The actual proximity of the closest principal stress to the vertical axis is indicated in Figure 8–28. The neighbouring node (334) and the middle injection node spend about 100 days each with the least principal stress rotated at approximately 45° to the vertical.

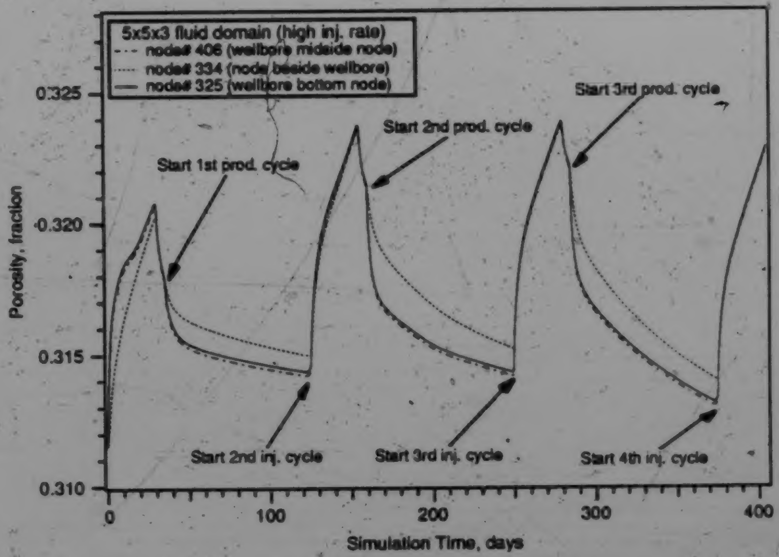


Figure 8–29

Change in porosity, ϕ, with simulation time
at selected nodes near the wellbore (5 × 5 × 3 fluid domain; high rate).

The porosities near the wellbore, presented in Figure 8–29, reflect only the changes in (pore) pressure. The unresolved stress state cannot affect the porosity in the uncoupled formulation. A coupled analysis, with porosity a function of volumetric strain would be different.

8.6 *Results for* $5 \times 5 \times 3$ *fluid domain with low injection rate*

Results are presented for the low-rate case (uncoupled) in the $5 \times 5 \times 3$ grid. Element and node numbers which are referred to are identified in Figure 8–5. Gauss point positions within elements are identified in Figure 8-7. The simulation took 111 time steps to reach 405 days of simulation time. The same number of time steps was used to find the solid model solution. The variation in pressure and temperature with simulation time at the selected nodes 325, 334, and 406 is shown in Figures 8–30 and 8–31, respectively. Nodes 325 and 406 are wellbore nodes, while node 334 is one of the nodes immediately adjacent to the bottom wellbore node, 325.

As expected, changes in pressure and temperature occur similarly to the high-rate case in the same grid, but with lower peaks due to the smaller slug of injected steam. As before, material behaviour beside these three nodes will be presented. Three Gauss points from elements 257, 321, and 265 (see Figure 8–5) were chosen such that they were the ones closest to the nodes to the nodes reported in Figures 8–30 and 8–31, respectively. The data points are organized in the next five diagrams so as to correspond to the nodes considered for the pressure and temperature profiles. The nodes and their closest Gauss points will be referred to interchangeably in the following text. The same terms reported for the previous fluid domain studies will be presented in this section.

Figure 8–32 illustrates the growth in plastic strain at the three Gauss points. The trend follows the injection cycles closely, although plastic straining does not begin until the simulation is into the second injection cycle. The bottom well node shows the greatest degree of plastic strain, until the onset of the fourth injection cycle when the values at the neighbouring node 265 become larger. As will be seen later, insufficient iteration to resolve the excess stresses becomes a problem in the third injection cycle. The error is not sufficient to lead to large (erroneous) plastic strains until late in the simulation, after the start of the fourth injection period. The finer mesh appears to give a lower overall plastic strain than for the $3 \times 3 \times 3$ case, where the effective plastic strain is larger in the bottom injection element, and where there is a bigger gap between the effective plastic strain in the middle and bottom elements. This is probably due to the more even distribution of thermal and pore pressure loads from the fluid model.

Figure 8–33 illustrates the variation in volumetric strain as injection continues. This responds in the expected fashion until about half way through the third injection cycle, where a sudden decrease in volumetric strain in the bottom element 257 is associated with the unresolved high stresses.

Figure 8–34 illustrates the change in the stress ratio at each of the selected Gauss points. As

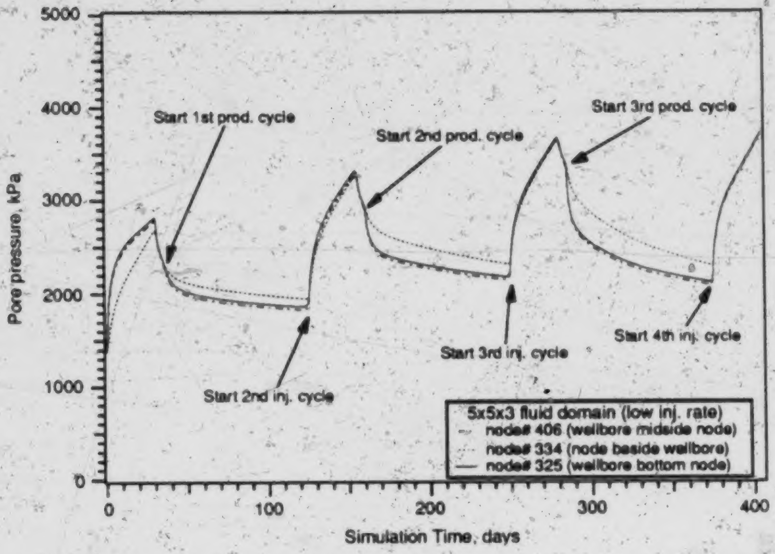


Figure 8–30

Profile of pressure (average block values) with simulation time at selected nodes near wellbore (5 × 5 × 3 fluid domain; low rate).

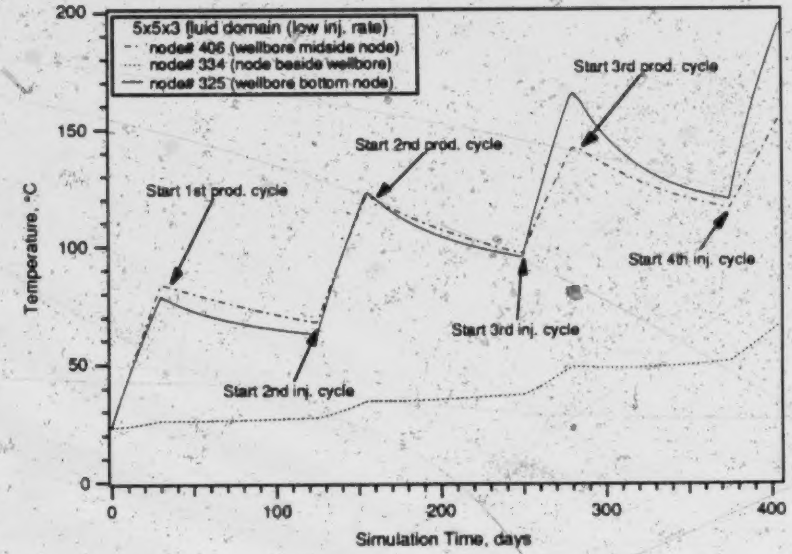


Figure 8–31

Profile of temperature (average block values) with simulation time at selected nodes near wellbore (5 × 5 × 3 fluid domain; low rate).

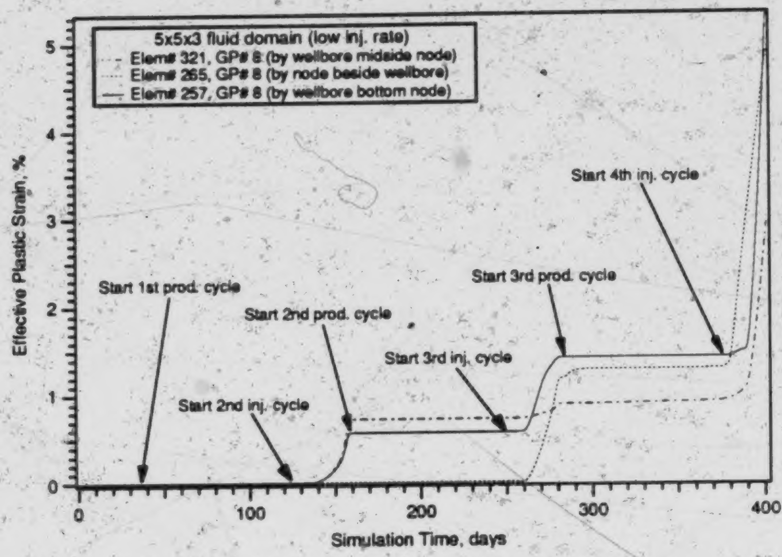


Figure 8–32

Growth of effective plastic strain with simulation time at selected Gauss points near the wellbore (5 × 5 × 3 fluid domain; low rate).

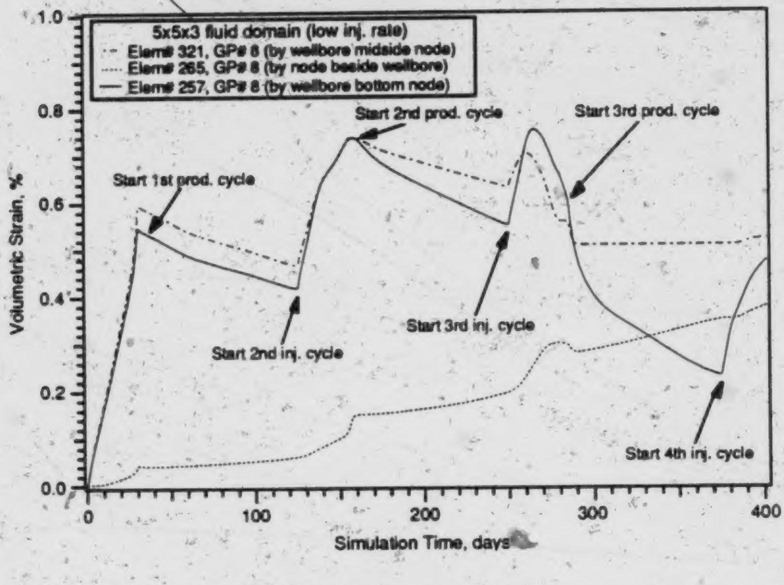


Figure 8–33

Growth of volumetric strain with simulation time at selected Gauss points near the wellbore (5 × 5 × 3 fluid domain; low rate).

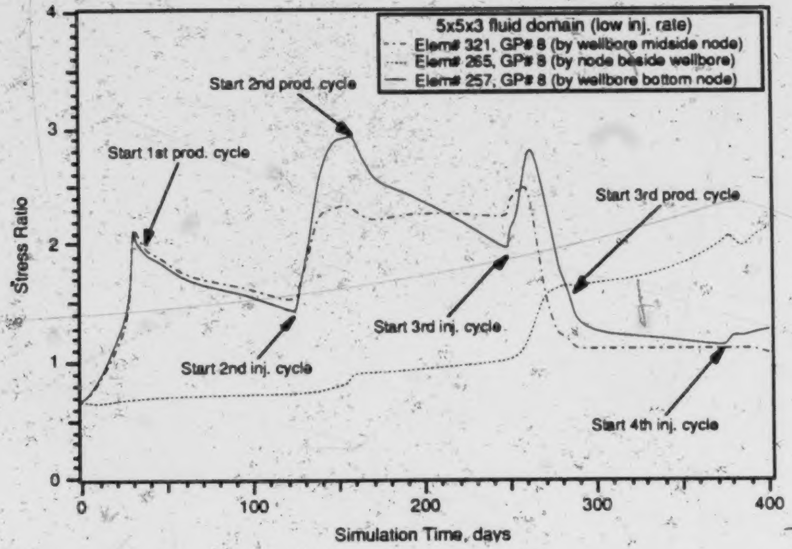


Figure 8–34

Change in stress ratio, K_o, with simulation time at selected Gauss points near the wellbore (5 × 5 × 3 fluid domain; low rate).

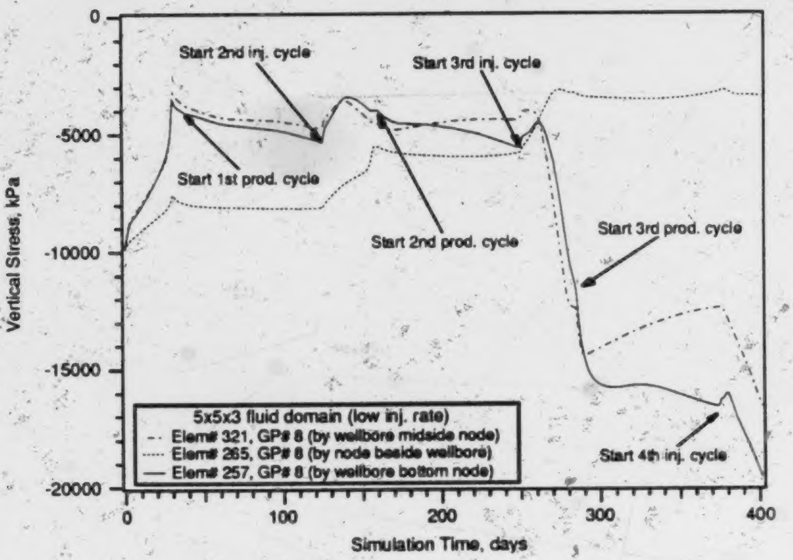


Figure 8–35

Change in vertical stress with simulation time at selected Gauss points near the wellbore (5 × 5 × 3 fluid domain; low rate).

before, the nodes at the wellbore quickly move to values of K_o above one during the first injection cycle, indicating that a horizontal orientation would be preferred in the event of a (tensile) hydraulic fracture. As in the high-rate case in the same mesh, the neighbouring Gauss point, in element 265, differs from the $3 \times 3 \times 3$ mesh results in that the stress ratio does not go above one until the third injection cycle. As the solid model results are satisfactory until the third injection cycle, the data to that point yield some insight. The high peak at the end of the first injection period reflects the higher well block temperatures caused by a smaller block than for the $3 \times 3 \times 3$ mesh.

The decrease in the stress ratio occurring in the wellbore region during the third injection period corresponds to the increase in vertical stress at the same time. This is illustrated in Figure 8–35, and is clearly seen at about 250 days. Again, this change is attributed to the unresolved stress state.

The next illustration, Figure 8–36, shows the Lode angle, θ, varying with time. It clearly illustrates the rotation of the principal stresses during the first injection cycle by moving from +30° to −30° very quickly. The unresolved stress state starts in the middle of the third injection period, and the Lode angle once again moves to values close to zero.

On the same page, Figure 8–37 illustrates the variation of minor principal effective stress (in the compression-positive sense) with time at the same selected nodes as the earlier pressure and temperature profiles. These values, as before, were obtained from stresses approximated at the nodes using the scheme described in §6.5.2. The results show that, unlike in the high-rate case, a tensile stress state is not achieved in the first injection period. The high saw-tooth values of the extrapolated least principal stress, beginning in the second injection cycle, are probably due only to the presence of large stress gradients in the bottom element, exaggerated by linear extrapolation to the nodes.

The principal stress numbers closest to the vertical, and the angles between them and the vertical, are given in Figures 8–38 and 8–39. The middle element Gauss point indicates that, during the third production period, the major principal stress is closest to the vertical. The cosine data also indicates that the same principal stress is inclined at 45° to the vertical at this point.

The porosities near the wellbore, presented in Figure 8–40, again, reflect only the changes in (pore) pressure. The unresolved stress state late in the simulation cannot affect the porosity in the uncoupled formulation.

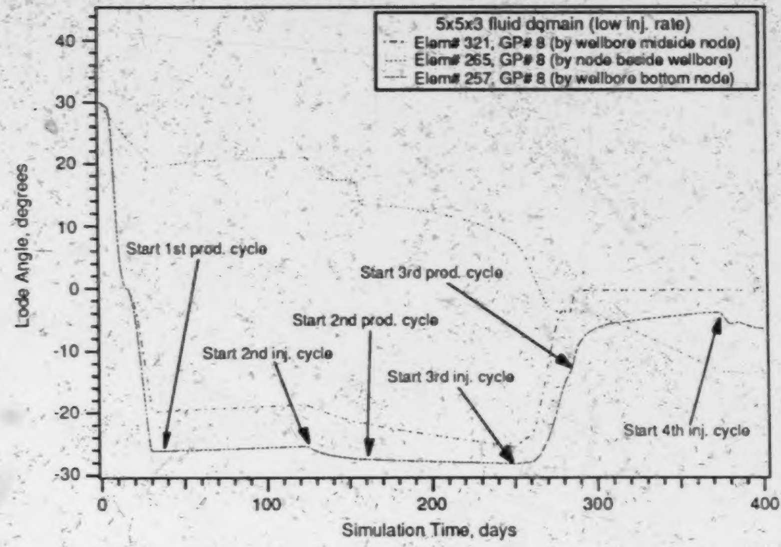


Figure 8–36

Change in Lode angle, θ, with simulation time at selected Gauss points near the wellbore ($5 \times 5 \times 3$ fluid domain; low rate).

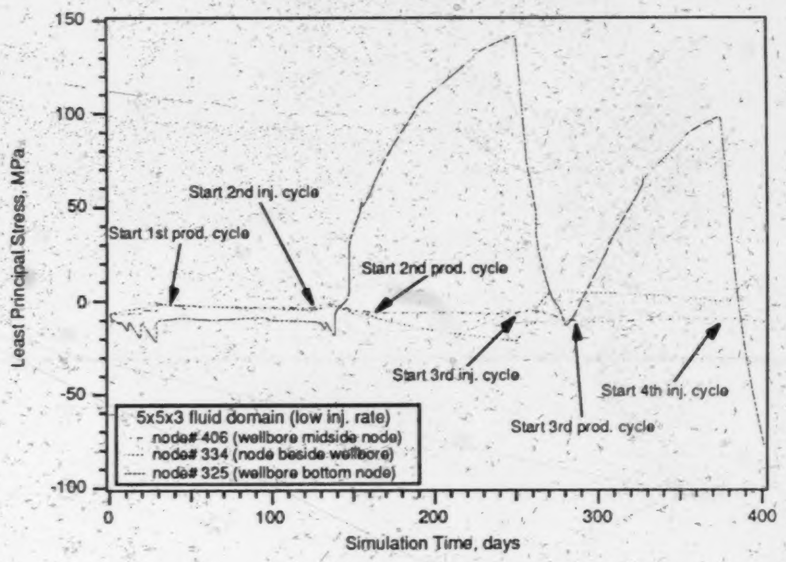


Figure 8–37

Change in least principal stress, σ_3, with simulation time at selected nodes near the wellbore ($5 \times 5 \times 3$ fluid domain; low rate).

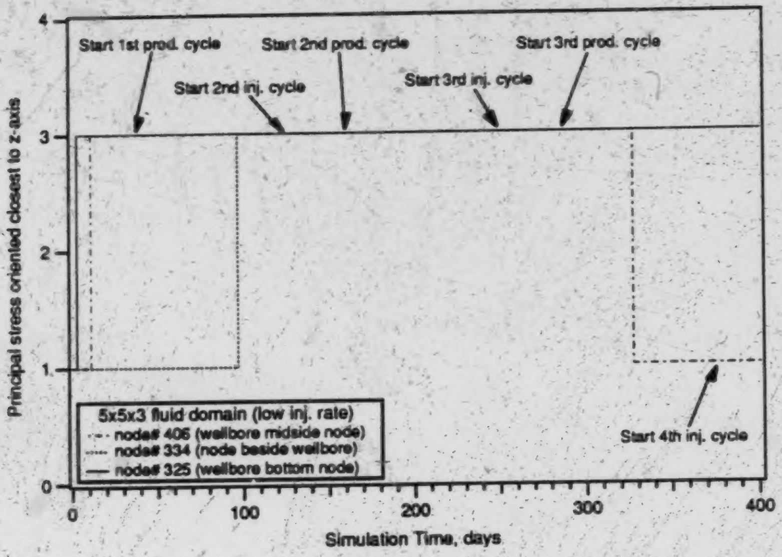


Figure 8–38

Change in the principal stress number oriented closest to the *z*-axis at selected nodes near the wellbore (5 × 5 × 3 fluid domain; low rate).

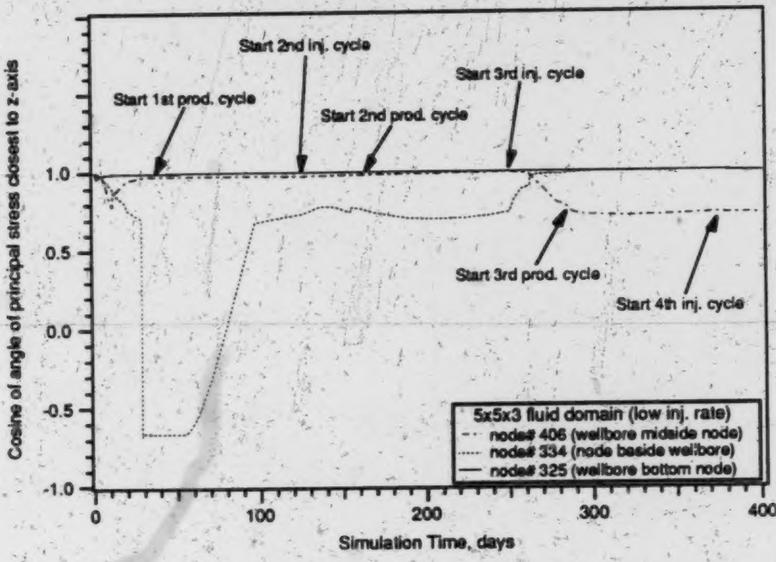


Figure 8–39

Change in the angle of the principal stress oriented closest to the *z*-axis at selected nodes near the wellbore (5 × 5 × 3 fluid domain; low rate).

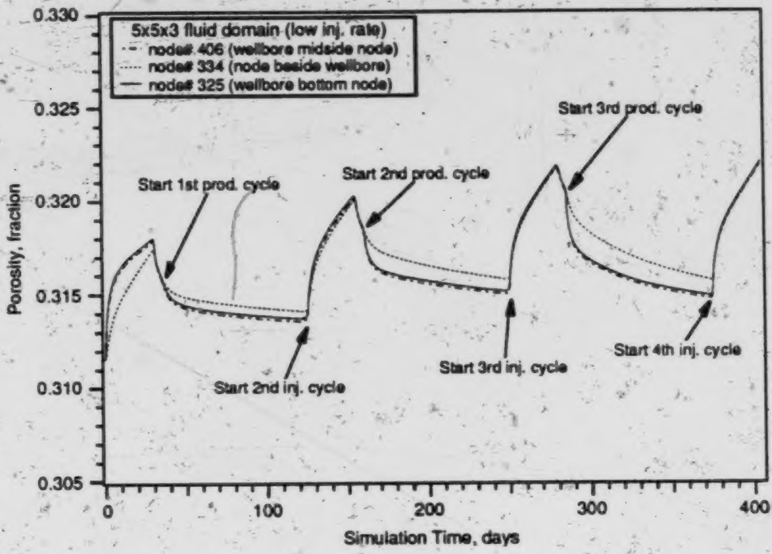


Figure 8–40

Change in porosity, ϕ, with simulation time at selected nodes near the wellbore ($5 \times 5 \times 3$ fluid domain; low rate).

8.7 *Results for 3 × 3 × 3 fluid domain with high injection rate and staggered coupling*

Results are presented for the high-rate case in the 3 × 3 × 3 grid, using the staggered coupling with the pore compressibility formulation. Element and node numbers which are referred to are identified in Figure 8–3. Gauss point positions within elements are identified in Figure 8–7. The simulation took 106 time steps to reach 254 days of simulation time (two full steam injection cycles). The variation in pressure and temperature with simulation time at the selected nodes 257, 265, and 321 is shown in Figures 8–41 and 8–42, respectively. Nodes 257 and 321 are wellbore nodes, while node 265 is one of the nodes immediately adjacent to the bottom wellbore node, 257.

These results should be compared to the uncoupled 3 × 3 × 3 case, Figures (8–8) to (8–18). The peak pressures at the end of the first injection period are higher, due to the different effective pore compressibilities. More interestingly, the pressures during the second production cycle do not display the same drop seen in the uncoupled runs. The pressure maintenance begins at about 150 days, following a short decline during the soak period. Consideration of other results, such as volumetric strain, vertical stresses, and the porosities, suggests that this phenomenon is explained by an increase in stress which compacts the formation. The increase in stress occurs at the well injection nodes, and is the result of unresolved excess stresses. However, there is no evidence that unresolved excess stresses exist at the element neighbouring the well injection element, and Figure 8–41 clearly indicates that the pressure in the neighbouring element is also maintained quite high during the second production cycle. This is partly be due to the pressure response in the well elements, through which a pressure gradient must be maintained for flow (hence the higher pressure than in the well elements). It is also observed that the volumetric strain in the neighbouring element is steadily increasing during the same period, a result of changing pore pressure and temperature distribution during this period. The temperature response is similar in form to the uncoupled case, as is expected—the temperature distribution will only be altered by changes in flow paths, or by the retention of hot fluids when flow is restricted. It is then noted that the temperatures in this case are all higher than in the uncoupled case, probably due to less convection of heat into the formation because of the different pressure response.

As before, material behaviour beside these three nodes will be presented. Three Gauss points from elements 197, 204, and 246 (see Figure 8–3) were chosen such that they were the ones closest to the nodes reported in Figures 8–41 and 8–42. The data points are organized in the next five diagrams so as to correspond to the nodes considered for the pressure and temperature profiles. The nodes and their closest Gauss points will be referred to interchangeably in the following text. The same terms reported for the previous fluid domain studies will be presented in this section.

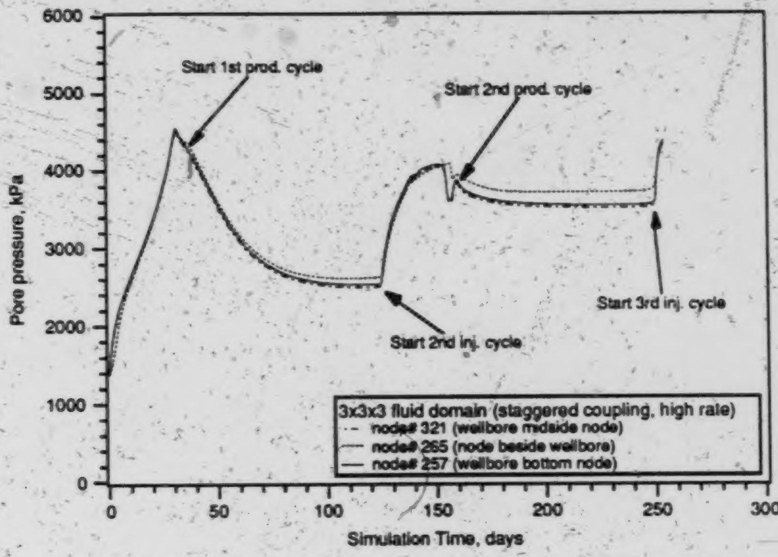


Figure 8–41

Profile of pressure (average block values) with simulation time at selected nodes near wellbore (3 × 3 × 3 fluid domain; high rate).

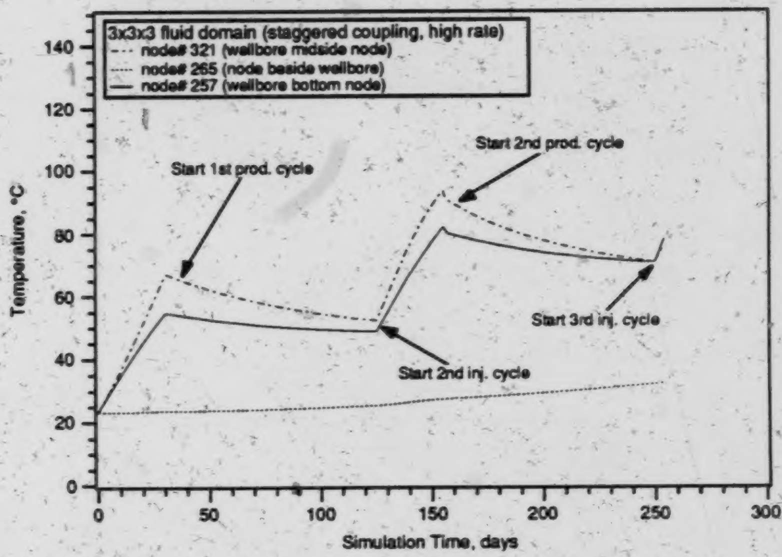


Figure 8–42

Profile of temperature (average block values) with simulation time at selected nodes near wellbore (3 × 3 × 3 fluid domain; high rate).

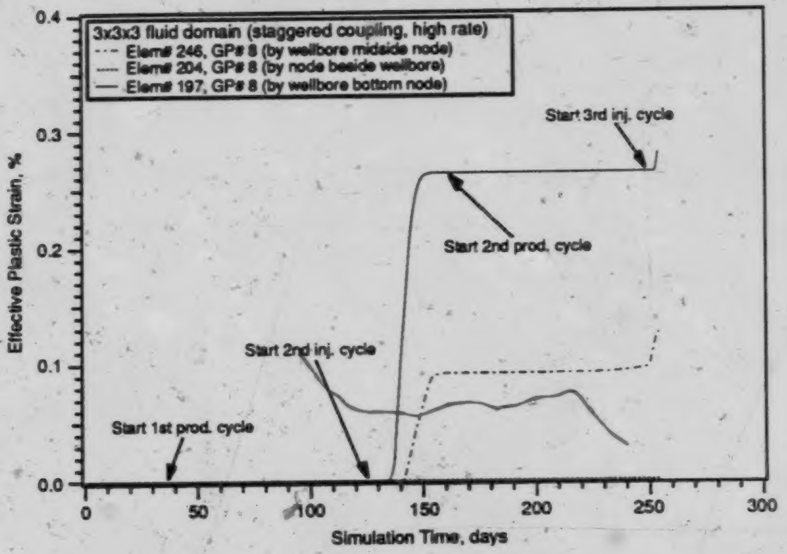


Figure 8–43

Growth of effective plastic strain with simulation time at selected Gauss points near the wellbore (3 × 3 × 3 fluid domain; high rate).

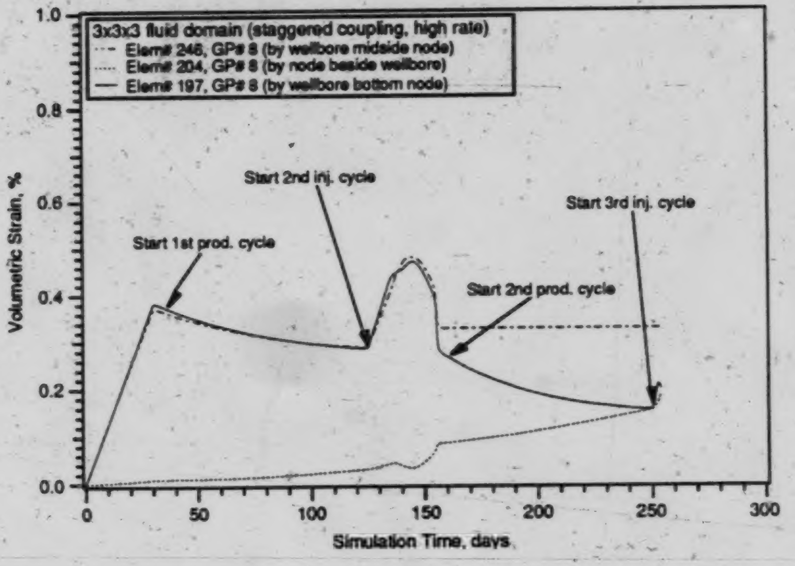


Figure 8–44

Growth of volumetric strain with simulation time at selected Gauss points near the wellbore (3 × 3 × 3 fluid domain; high rate).

Figure 8–43 illustrates the growth in plastic strain at the three Gauss points. The trend follows the injection cycles closely, although plastic straining does not begin until the simulation is into the second injection cycle. The bottom well Gauss point shows the greatest degree of plastic strain. The values at the neighbouring element Gauss point 204 are zero throughout the time of the simulation, indicating that the response at that node is purely elastic. All the values are similar in magnitude to those seen in the uncoupled case. The inadequate resolution of the excess stresses back to the yield surface does not significantly affect the effective plastic strain, as it was mostly accumulated during the injection period.

Figure 8–44 illustrates the variation in volumetric strain as injection continues. The response mirrors the variation in vertical stress reported later. The sudden drop in ϵ_v half way through the second injection period corresponds to the sudden increases in vertical stress at the same time. However, the response during the second production period is more varied. The volumetric strain continues to decline in the bottom well element, reflecting the continued but more gentle decline in vertical stress (an increase in its magnitude). The volumetric strain at the middle well element remains more or less constant, while the value at the neighbouring element 204 increases steadily.

Figure 8–45 illustrates the change in the stress ratio at each of the selected Gauss points. The sharp decline in stress ratio during the second injection period reflects the increasing vertical stress magnitude at the same time, due to insufficient reduction of the excess stresses. The values stay greater than one at all locations, once the value of one has been attained. The first peak in K_o at the end of the first injection period is unlike that of the uncoupled case with the same mesh, but is not unlike that in the high-rate $5 \times 5 \times 3$ case. This reflects the higher temperatures in the well elements, discussed earlier.

The vertical stress response, illustrated in Figure 8–46, has already been discussed in conjunction with the previous parameters.

The Lode angle, θ, shown in Figure 8–47, follows a similar path to that in the uncoupled runs. The values at the points close to the injection nodes indicate principal stress rotation occurring during the first injection period. The neighbouring point in element 204 moves more slowly towards the rotated state. The two points by the injection nodes move to values close to zero about half way through the second injection period, further evidence of unresolved excess stresses.

On the same page, Figure 8–48 illustrates the variation of minor principal effective stress (in the compression-positive sense) with time at the same selected nodes as the earlier pressure and temperature profiles. These values, as before, were obtained from stresses approximated at the

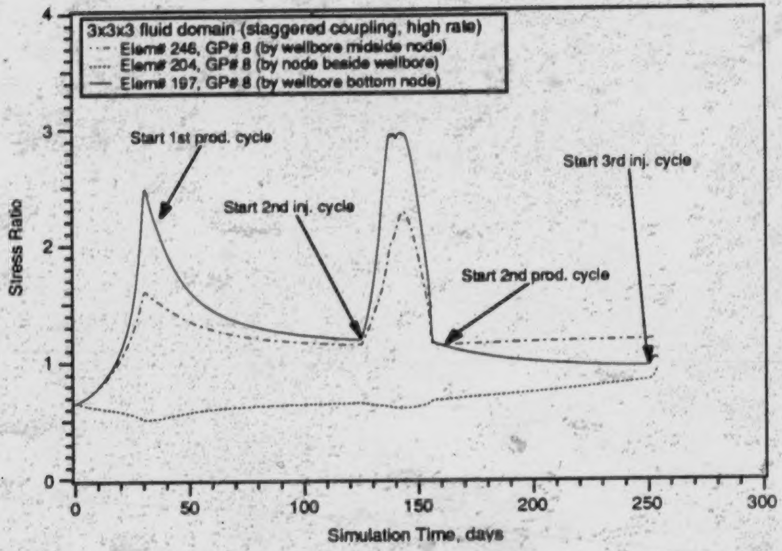


Figure 8–45

Change in stress ratio, K_o, with simulation time at selected Gauss points near the wellbore (3 × 3 × 3 fluid domain; high rate).

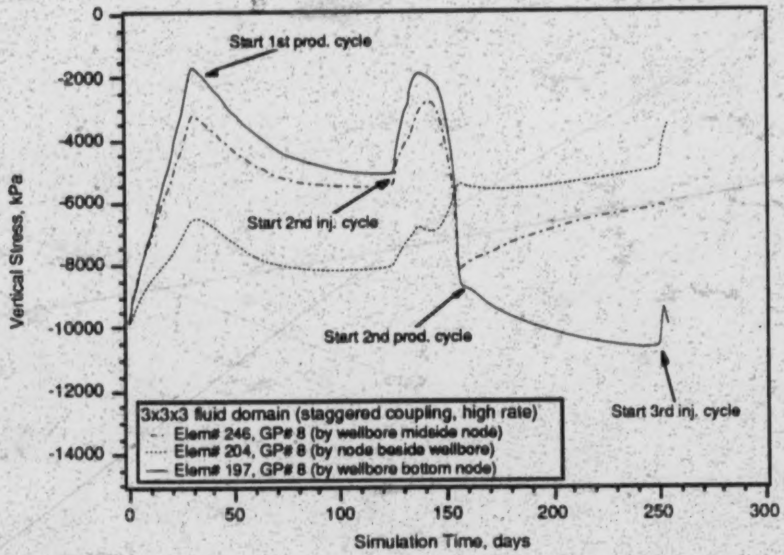


Figure 8–46

Change in vertical stress with simulation time at selected Gauss points near the wellbore (3 × 3 × 3 fluid domain; high rate).

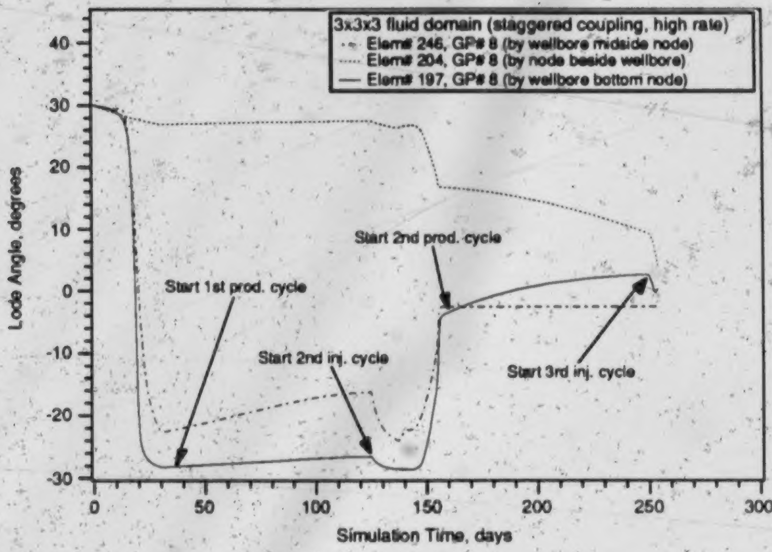


Figure 8–47

Change in Lode angle, θ, with simulation time at selected Gauss points near the wellbore ($3 \times 3 \times 3$ fluid domain; high rate).

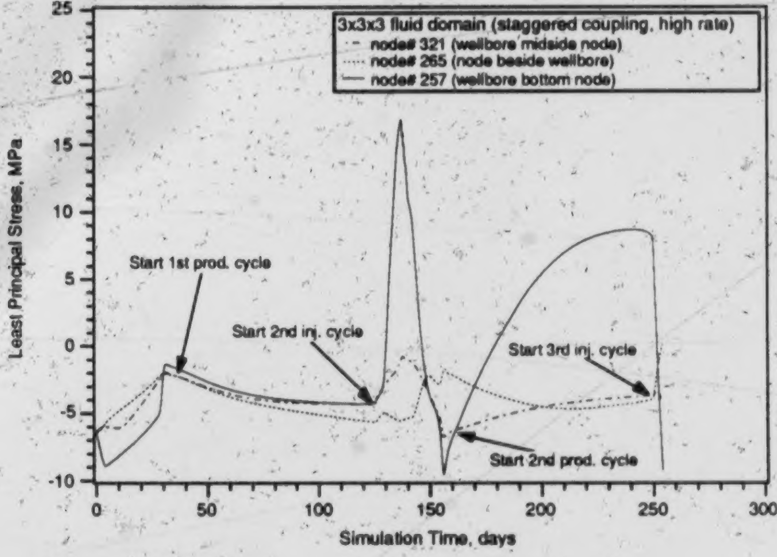


Figure 8–48

Change in least principal stress, σ_3, with simulation time at selected nodes near the wellbore ($3 \times 3 \times 3$ fluid domain; high rate).

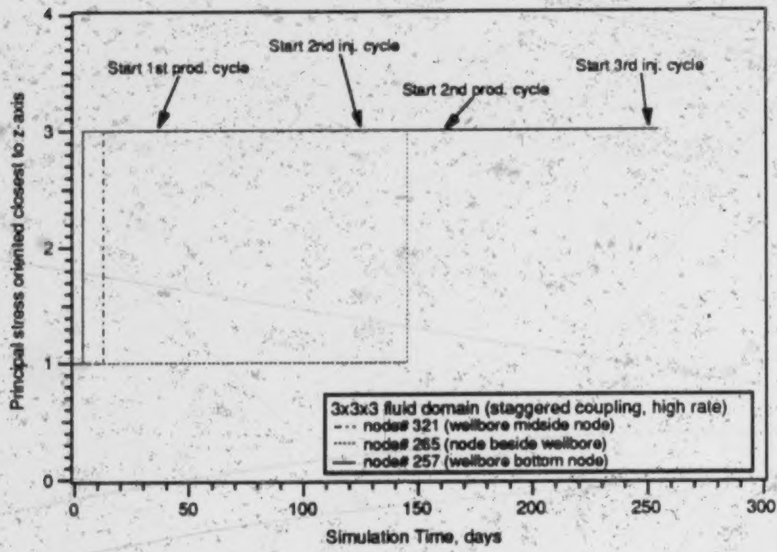


Figure 8–49

Change in the principal stress number oriented closest to the z-axis at selected nodes near the wellbore (3 × 3 × 3 fluid domain; high rate).

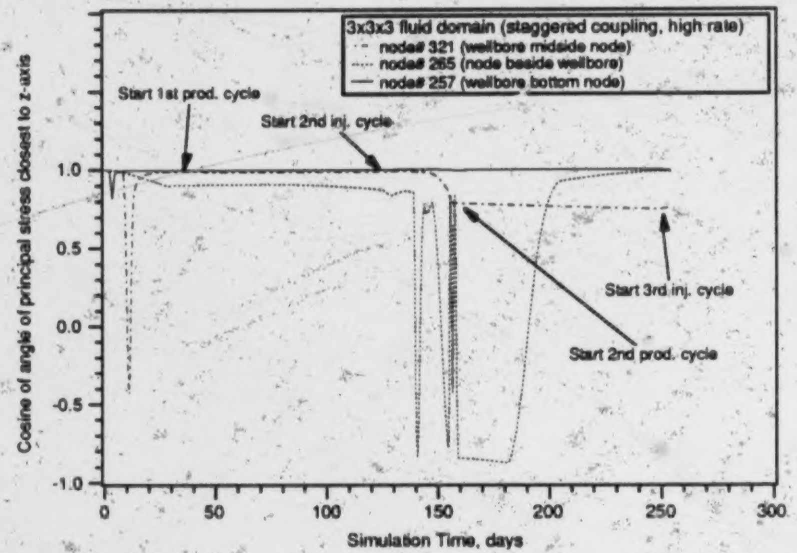


Figure 8–50

Change in the angle of the principal stress oriented closest to the z-axis at selected nodes near the wellbore (3 × 3 × 3 fluid domain; high rate).

nodes using the scheme described in §6.5.2. The values for the bottom well node become anomalous at 140 days, with a large positive spike followed by oscillation to extremes. This is a result of the poor reduction of excess stress in that element. The values at the middle injection element are not so severe, although the increase (in magnitude) at 140 days is also a sign that some excess stress still exists. Excluding the anomalous values at node 257, the least principal stresses plotted do not go into tension during the simulation period.

The principal stress numbers closest to the vertical, and the angles between them and the vertical, are given in Figures 8–49 and 8–50. Principal stress rotation occurs in the order discussed with the Lode angle. At all times, the principal stress at the bottom wellbore node 257 coincides with the vertical axis. The middle well node deviates from this about half way through the second injection period, moving to the angle of 45° to the vertical. Node 265, neighbouring the bottom well node, also moves to the 45° angle in the second injection cycle. This might be a response to the excess stress in the wellbore elements.

In the coupled runs, the porosities are calculated directly from the volumetric strains and the thermal expansions. Thus the values reflect directly the interaction between the fluid and solid models. As is expected, the porosity values, presented in Figure 8–51, have a similar profile to the volumetric strain curves, with some exaggeration of the differences between the nodes. The permeabilities, related by a cubic expression to the volumetric strains and porosities, change at the selected nodes as shown in Figure 8–52. The relative shape of each curve is closest to the variation in porosity, and reaches a high point of four during the second injection cycle. This has a corresponding effect on the block transmissibilities, and on the injection pressures.

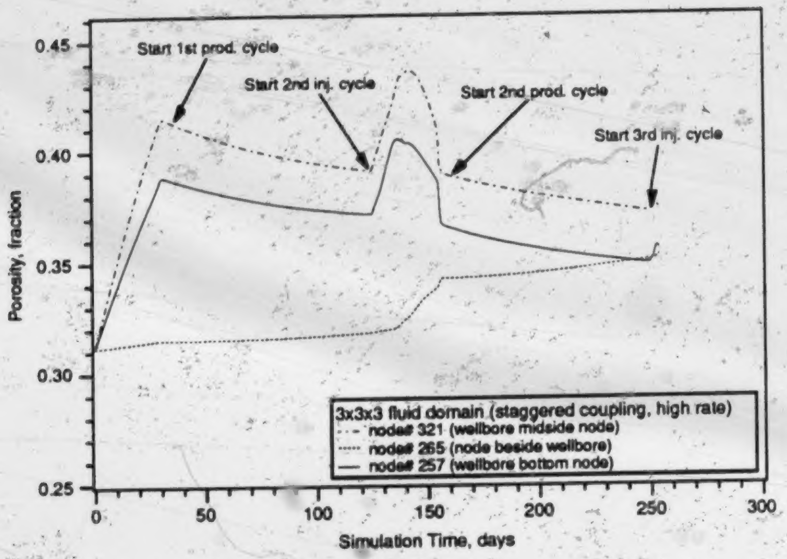


Figure 8–51

Change in porosity, ϕ, with simulation time at selected nodes near the wellbore ($3 \times 3 \times 3$ fluid domain; high rate).

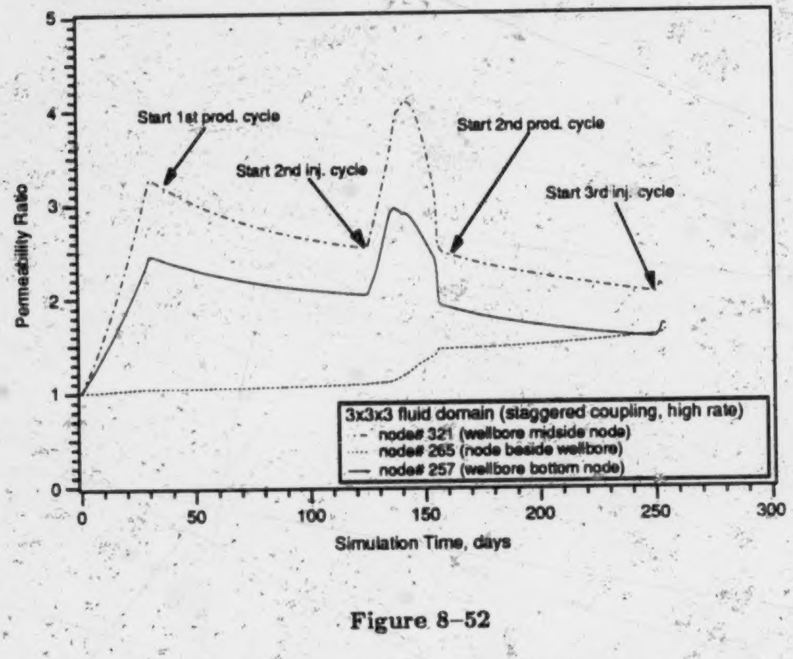


Figure 8–52

Change in permeability, k, with simulation time at selected nodes near the wellbore ($3 \times 3 \times 3$ fluid domain; high rate).

8.8 *Results for* 3 × 3 × 3 *fluid domain with low injection rate and full coupling*

Results are presented for the low-rate case in the 3 × 3 × 3 grid, using the full coupling. Element and node numbers which are referred to are identified in Figure 8–3. Gauss point positions within elements are identified in Figure 8–7. The simulation took 85 time steps to reach 106 days of simulation time (one full steam injection cycle). The solution diverged at this time, and further progress was not possible. The variation in pressure and temperature with simulation time at the selected nodes 257, 265, and 321 is shown in Figures 8–53 and 8–54, respectively. Nodes 257 and 321 are wellbore nodes, while node 265 is one of the nodes immediately adjacent to the bottom wellbore node, 257.

These results should be compared to the uncoupled 3 × 3 × 3 (high rate) case, and to the the staggered case (high rate). While the injection rate for this case is less than that for the other two coarse grid scenarios, one can still consider and interpret the progress of changes.

The peak pressure at the end of the first injection period is even more pronounced than for when the staggered coupling was used. This can be ascribed to the lower effective pore compressibility of the system. It is also noted that the difference in pressure between the neighbouring node 265 and the well nodes is much less in the two coupled runs than for the uncoupled cases. This can be explained by the increase in absolute permeability accompanying the injection of steam. There is also a greater drawdown at the well nodes than in the previous examples. Here the explanation also hinges on the values of volumetric strain, which will be seen to be smaller for this example. The low pressures at the end of the first production period probably contributed to the eventual divergence of the solution.

The temperature response is similar in form to the uncoupled case, as is expected—the temperature distribution will only be altered by changes in flow paths, or by the retention of hot fluids when flow is restricted. It is then noted that the temperatures in this case are all lower than in the uncoupled case, reflecting the lower injection rate used in the coupled run.

As before, material behaviour beside these three nodes will be presented. Three Gauss points from elements 197, 204, and 246 (see Figure 8–3) were chosen such that they were the ones closest to the nodes reported in Figures 8–53 and 8–54. The data points are organized in the next five diagrams so as to correspond to the nodes considered for the pressure and temperature profiles. The nodes and their closest Gauss points will be referred to interchangeably in the following text. The same terms reported for the previous fluid domain studies will be presented in this section.

No effective plastic strain was noted in the previous examples until the second injection period.

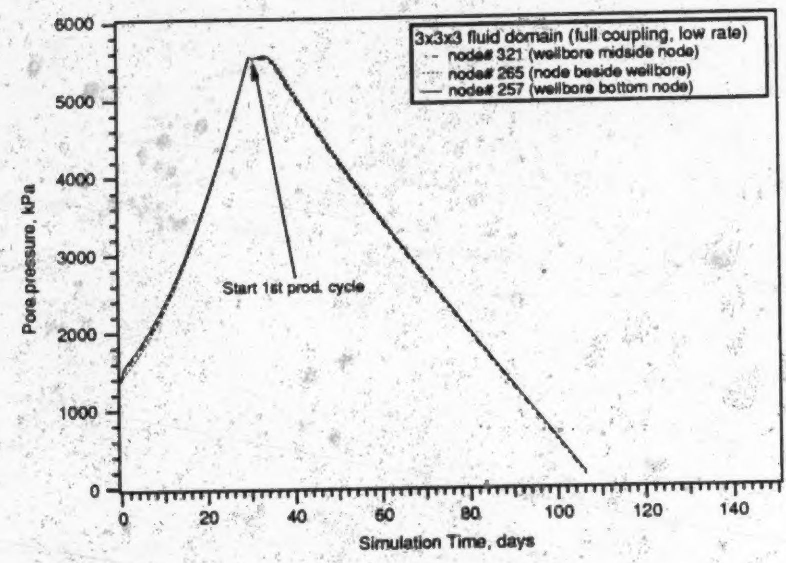


Figure 8–53

Profile of pressure (average block values) with simulation time at selected nodes near wellbore (3 × 3 × 3 fluid domain; low rate).

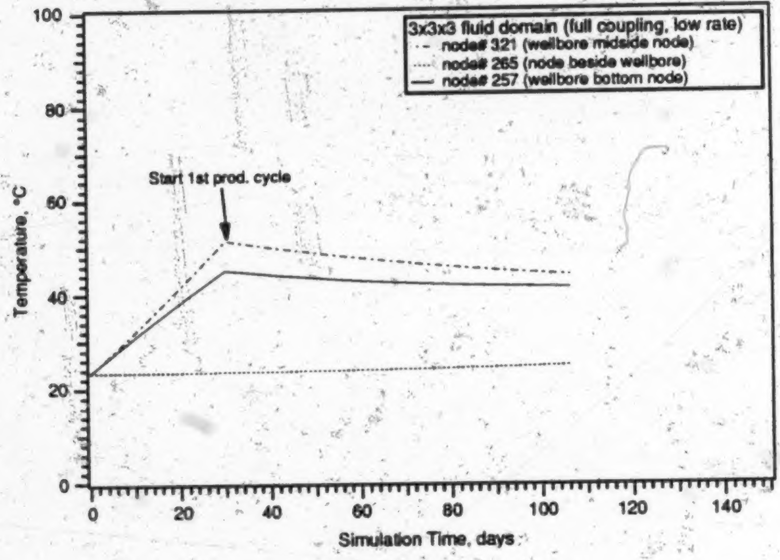


Figure 8–54

Profile of temperature (average block values) with simulation time at selected nodes near wellbore (3 × 3 × 3 fluid domain; low rate).

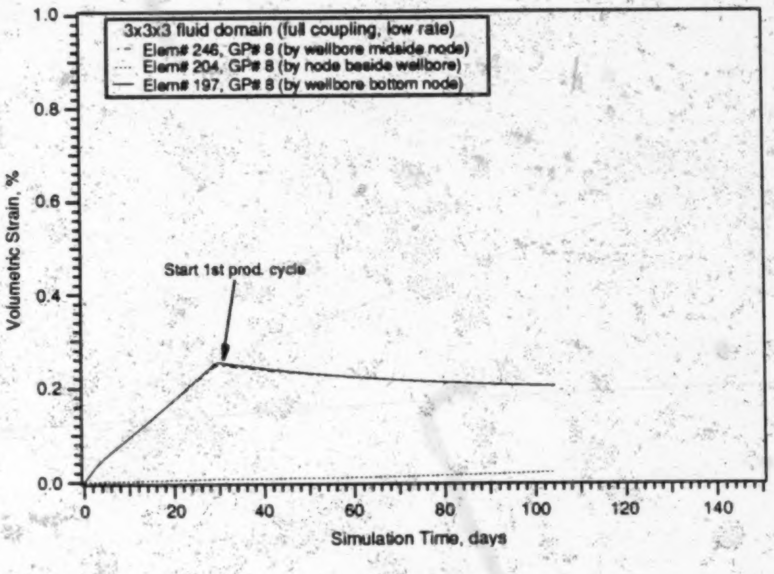


Figure 8–55

Growth of volumetric strain with simulation time at selected Gauss points near the wellbore (3 × 3 × 3 fluid domain; low rate).

As expected in a run terminated before entering the second injection period, no plastic strain was observed in this case. Figure 8–55 illustrates the variation in volumetric strain. It can be seen that the volumetric strain is lower than for the uncoupled case, and is mostly due to the lower injection rate used for this run.

Figure 8–56 illustrates the change in the stress ratio at each of the selected Gauss points. The two coupled runs display a different response at the neighbouring Gauss point (element 204), compared to the uncoupled run. In this run the difference is very noticable, where the stress ratio K_o actually decreases during the first injection period, in contrast to the uncoupled run where it increases slightly. The change in K_o can only happen as a result of a reduction in the major horizontal stress or an increase in the vertical stress. The vertical stress response, seen in Figure 8–57, shows that the vertical stress (magnitude) at this Gauss point decreases more than in the uncoupled runs. It is probable that the lower injection rate of the coupled run, resulting in a lower thermal loading, is the main reason for this behaviour.

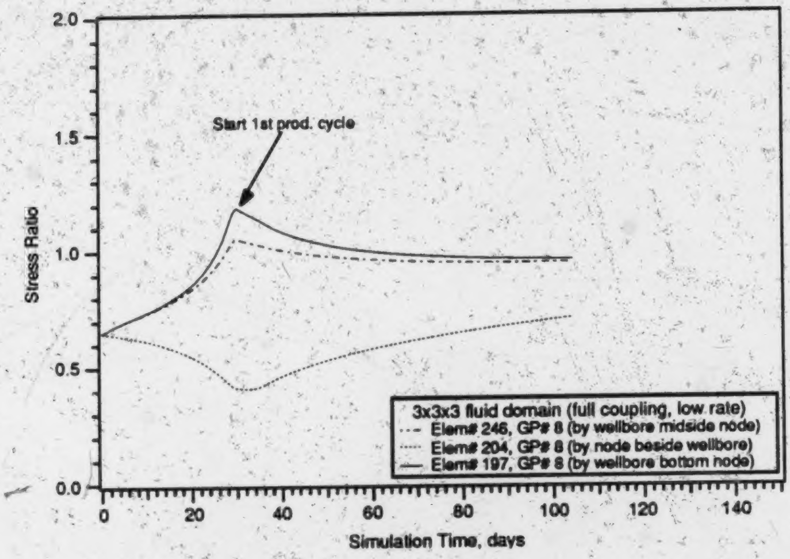


Figure 8–56

Change in stress ratio, K_o, with simulation time at selected Gauss points near the wellbore (3 × 3 × 3 fluid domain; low rate).

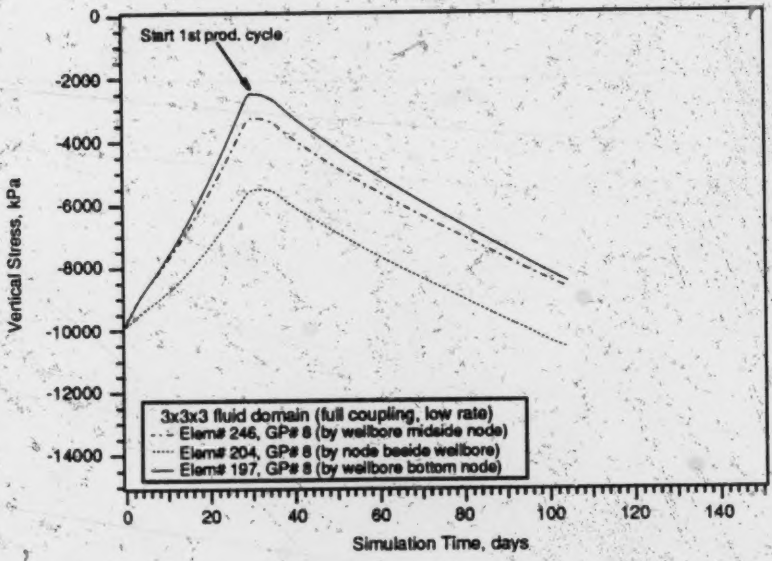


Figure 8–57

Change in vertical stress with simulation time at selected Gauss points near the wellbore (3 × 3 × 3 fluid domain; low rate).

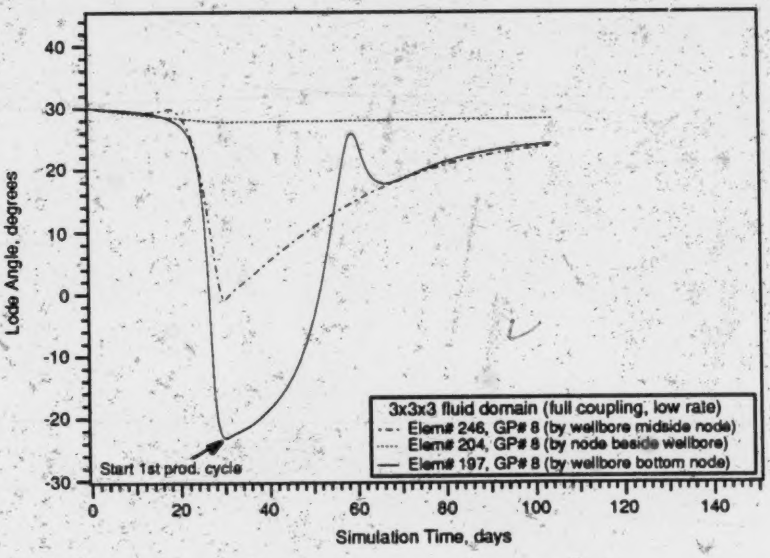


Figure 8–58

Change in Lode angle, θ, with simulation time at selected Gauss points near the wellbore ($3 \times 3 \times 3$ fluid domain; low rate).

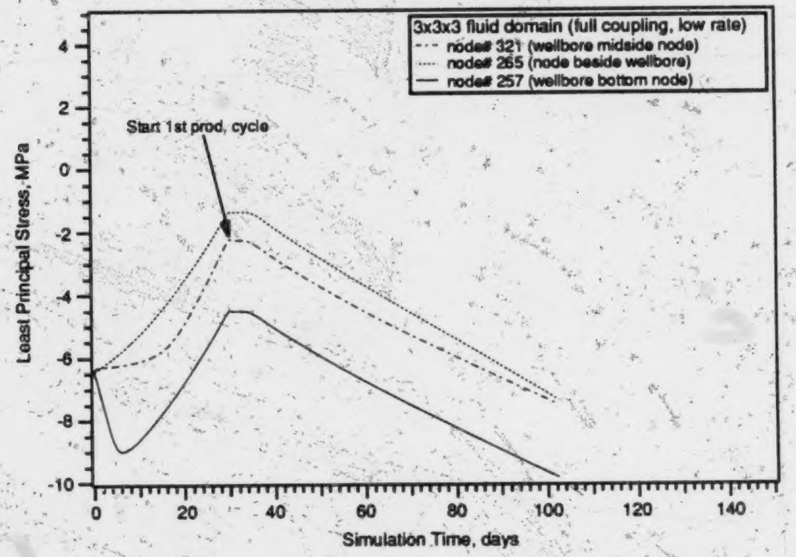


Figure 8–59

Change in least principal stress, σ_3, with simulation time at selected nodes near the wellbore ($3 \times 3 \times 3$ fluid domain; low rate).

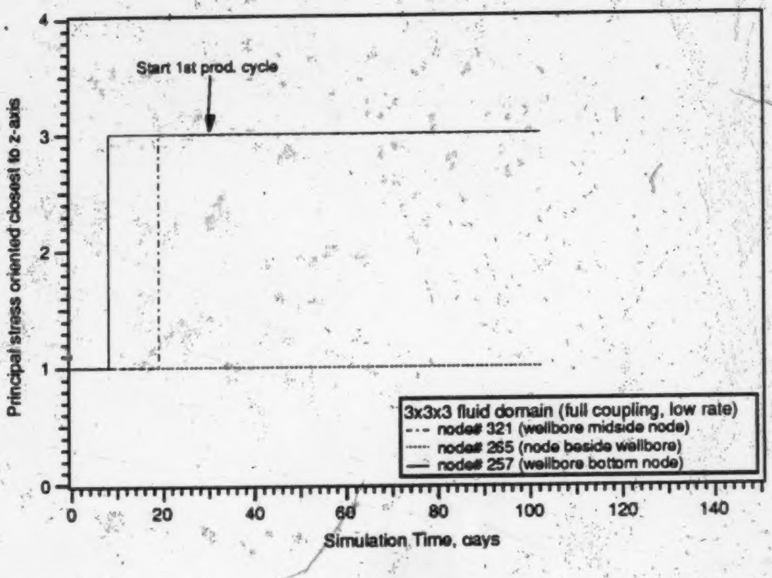


Figure 8–60

Change in the principal stress number oriented closest to the z-axis at selected nodes near the wellbore (3 × 3 × 3 fluid domain; low rate).

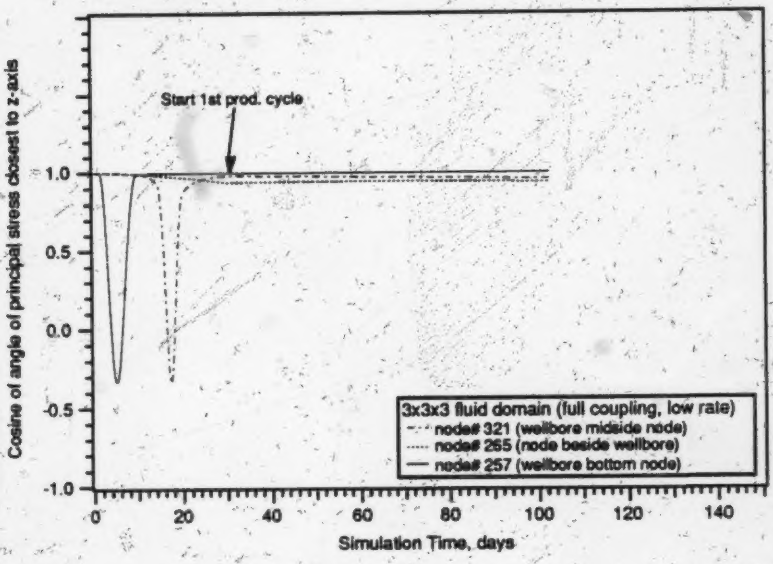


Figure 8–61

Change in the angle of the principal stress oriented closest to the z-axis at selected nodes near the wellbore (3 × 3 × 3 fluid domain; low rate).

The Lode angle, θ, shown in Figure 8–58, complements the changes in the stress ratio. Principal stress rotation, evidenced by a change in the Lode angle from +30° to −30°, occurs only at the Gauss points close to the well nodes used as an illustration. This is mirrored in the stress ratio figure, where the values at these points become greater than one. The stresses at the neighbouring point remain unrotated.

On the same page, Figure 8–59 illustrates the variation of minor principal effective stress (in the compression-positive sense) with time at the same selected nodes as the earlier pressure and temperature profiles. These values, as before, were obtained from stresses approximated at the nodes using the scheme described in §6.5.2. It is clearly seen in this diagram that the smallest minor principal stress (by magnitude) is at the neighbouring point, rather than directly adjacent to the well nodes. Using the information given in Figures 8–60 and 8–61, the principal stress numbers closest to the vertical, and the angles between them and the vertical, it can be seen that the neighbouring point has the major principal stress most closely oriented to the vertical axis, and that it is almost vertical. A vertical parting plane is then the preferred orientation if the point should attain zero effective stress.

The porosity values, presented in Figure 8–62, and the permeability ratios, shown in Figure 8–63, have a similar profile to the volumetric strain curves, again with some exaggeration of the differences between the nodes. These variations, as a result of the volumetric strain response, alter the response of the reservoir to steam injection compared to the uncoupled analysis.

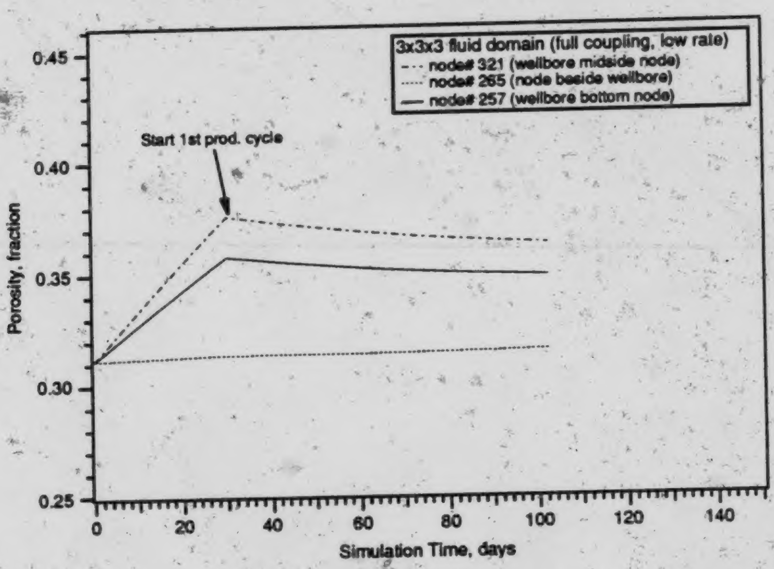


Figure 8–62

Change in porosity, ϕ, with simulation time

at selected nodes near the wellbore (3 × 3 × 3 fluid domain; low rate).

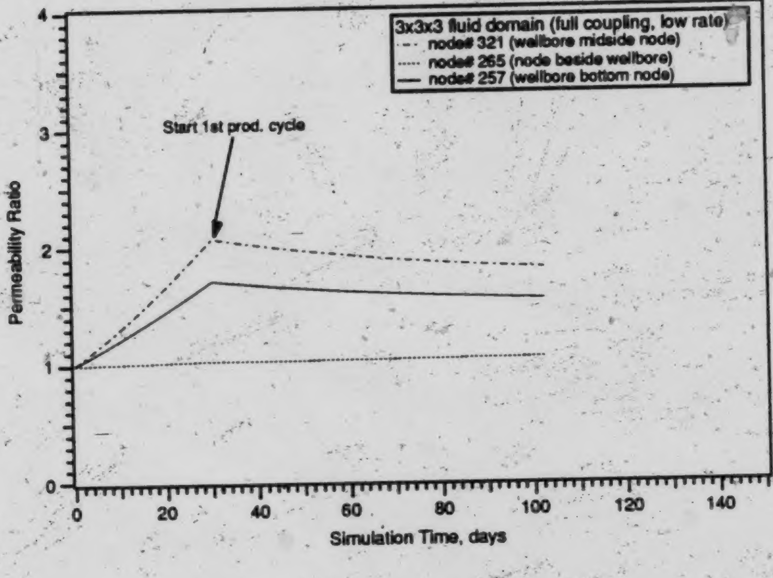


Figure 8–63

Change in permeability, k, with simulation time

at selected nodes near the wellbore (3 × 3 × 3 fluid domain; low rate).

8.9 *Discussion*

The numerical experiments chosen were intended to investigate the effects of changing important parameters on the differences encountered when using the coupled formulations, and to demonstrate the implementation of the coupled formulations. The most important result has been to demonstrate that the coupled runs are feasible, but are difficult and costly to perform. There are also stability problems, which were partly resolved, and the problem of obtaining stress solutions which had been reduced back to the yield surfaces from the excess trial values. As in most coupled problems, the amount of work to be done increases with the amount of coupling required. The staggered formulation is the simplest level of coupling, and if it gives realistic solutions, is the obvious candidate for further study.

8.10 *Comparison of methods*

Several individual parameters have been prepared in more detail for discussion: the production data, the plastic zone, the volumetric strain, and the vertical effective stresses. The production data is considered for all the runs. For the rest of the comparison, only the staggered coupled run (R333STAG) is compared with the two uncoupled runs operating at the higher rate (R333HR and R553HR). The contours for the 3 × 3 × 3 fluid model grid are taken from the bottom layer of Gauss points in the lower layer of elements in a 3 × 3 areal slice taken from the injection corner. The contours for the 5 × 5 × 3 fluid model grid are similarly taken from a 5 × 5 areal slice. Data is taken at 155 days, at the end of the second injection period for all the runs.

8.10.1 *Production data*

Some production parameters have been chosen to highlight differences and similarities between the simulations. The first diagram, Figure 8–64, illustrates the cumulative steam injection in barrels of cold water equivalent (cwe). All the wells were injection-rate specified, so unless a wellbore exceeded a specified pressure, the cumulative volumes injected should be the same between different simulations at the same injection rate. The two lines underscore this, and the difference between the two rates is clear.

Figure 8–65 illustrates the oil production rate with simulation time for each case. Production is confined to specified cycles, and the rate increases with each successive cycle for the three considered here. Of the two coupled runs, there is only enough data from the staggered coupling to be able to make any comparison with the uncoupled cases. The 3 × 3 × 3 runs have similar profiles during the production periods, but with some subtle differences. The finer 5 × 5 × 5 grid makes a more realistic production curve, with high initial rates and rapid declines during the course of the production period. The staggered run produces oil a little slower than the uncoupled counterpart in both the cycles for which there is data. Water production, shown in Figure 8–66, is higher in the (staggered) coupled model. Presumably the more mobile water can flow faster in the enhanced (absolute) permeability caused by the dilation, proportionately higher than the oil can. The mobile water is mostly steam condensate.

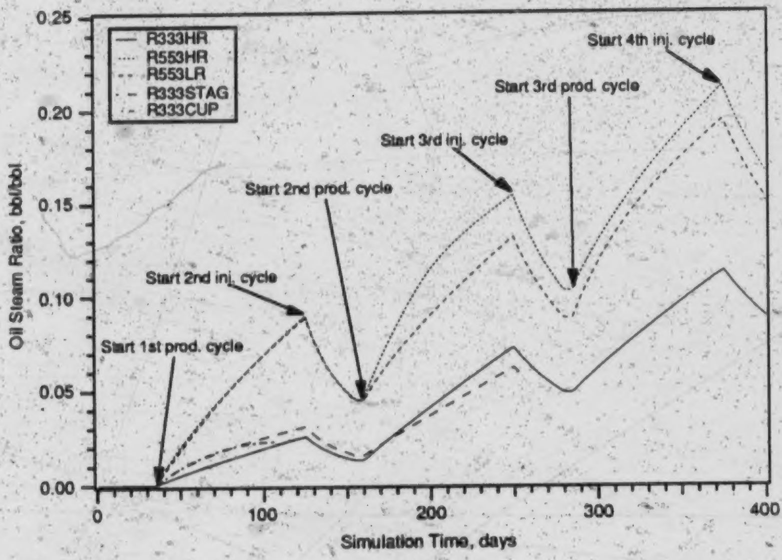


Figure 8–68

Profile of oil-steam ratio during simulations.

The cumulative oil production, shown in Figure 8–67 in % pore volumes, indicates that the staggered run only had a slightly different oil recovery compared to that from the uncoupled run. The fine mesh simulations give better recoveries than the others because of the different rates of mobilisation of bitumen between the finer and coarser grids. Given that the injection rates are constant, it is not surprising that the oil-steam ratios, plotted in Figure 8–68 reflect the figures for cumulative oil production.

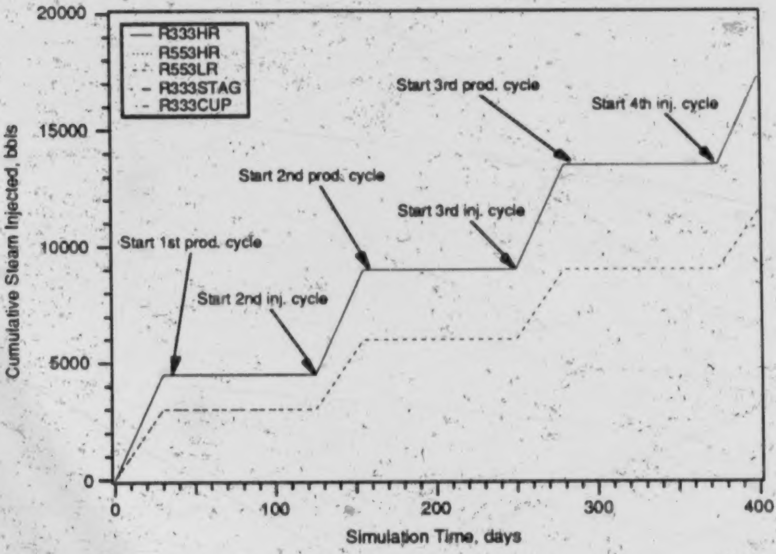


Figure 8–64

Profile of cumulative steam injection during simulations.

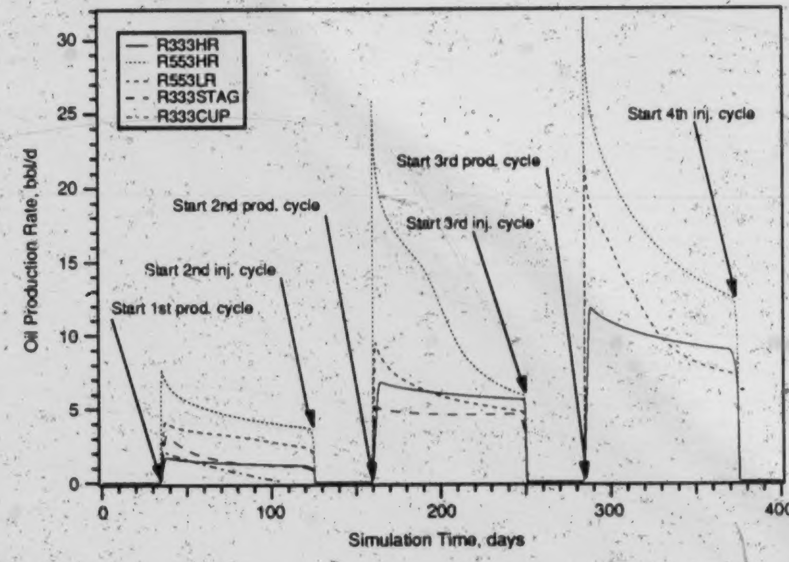


Figure 8–65

Profile of oil production rates during simulations.

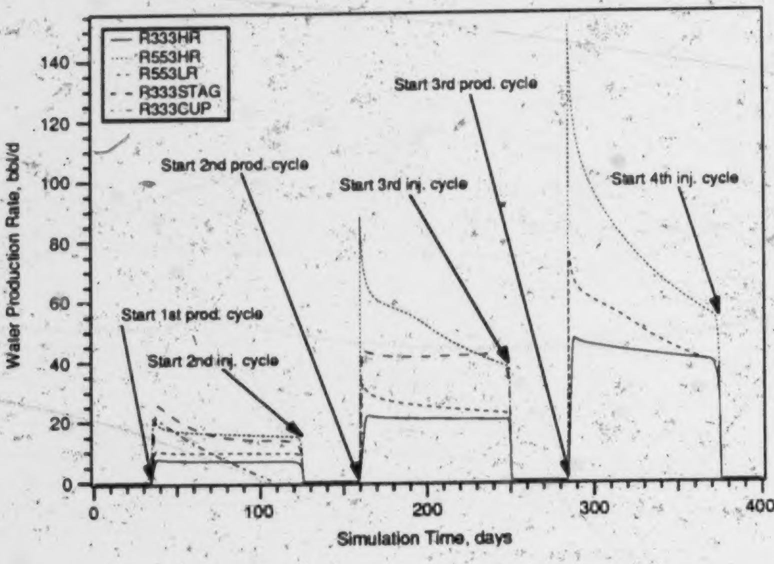


Figure 8–66

Profile of water production rate during simulations.

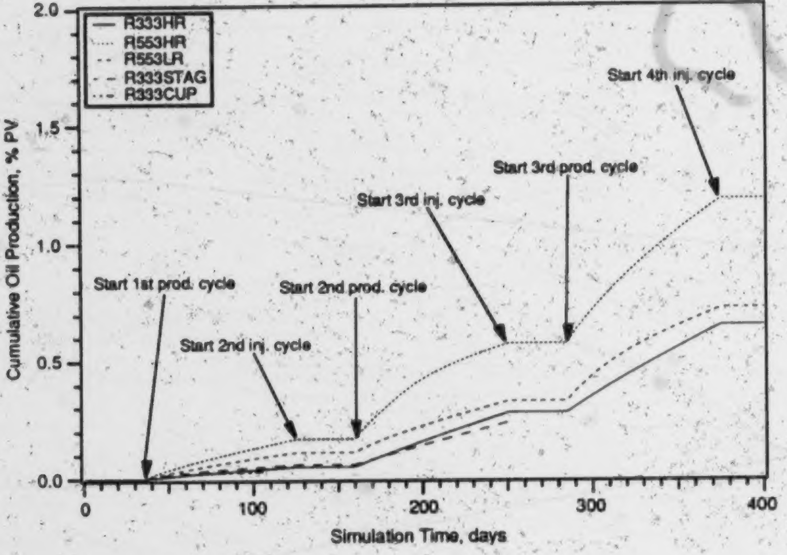


Figure 8–67

Profile of cumulative oil production during simulations.

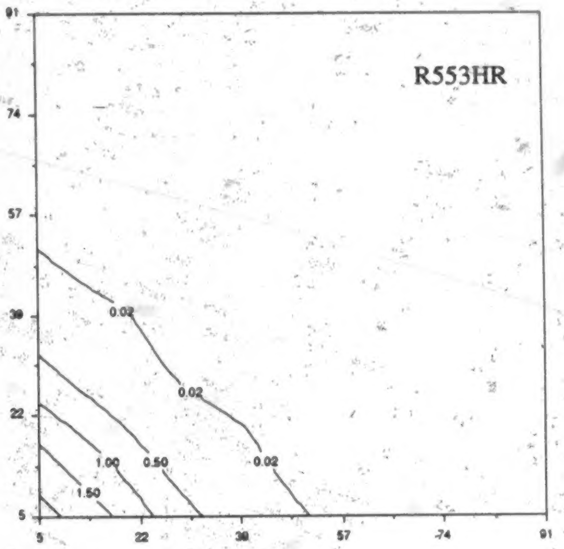


Figure 8–69

Contours of effective plastic strain (%) in R553HR from basal layer of Gauss points in heated zone.

8.10.2 *Comparison of plastic zone sizes*

The effective plastic strain is an indicator of plastic deformation, and therefore the boundary between zero and positive plastic strain is the boundary between plastic and elastic analysis, where the strains can accumulate differently from even non-linear elastic behaviour. In the two coarse grid simulations (both high rate, one with staggered coupling and one uncoupled), the only yielded Gauss point at this time and in the layer being considered, was the one nearest the well injection node. Only the contour map for the fine grid simulation (uncoupled, high rate) is presented in Figure 8–69. Yielding has occurred in more than one-quarter of the area, reflecting the better distribution of fluid pressure and temperature in the finer grid. A maximum value of 2.2% is at the corner Gauss point (contouring is based on a grid of the Gauss point data). The plastic zone is areally symmetrical.

8.10.3 *Comparison of volumetric strains*

The variation of volumetric strain with time impinges directly on the the fluid response, as porosity and permeability are functions of these strains in the coupled formulations. Contours at

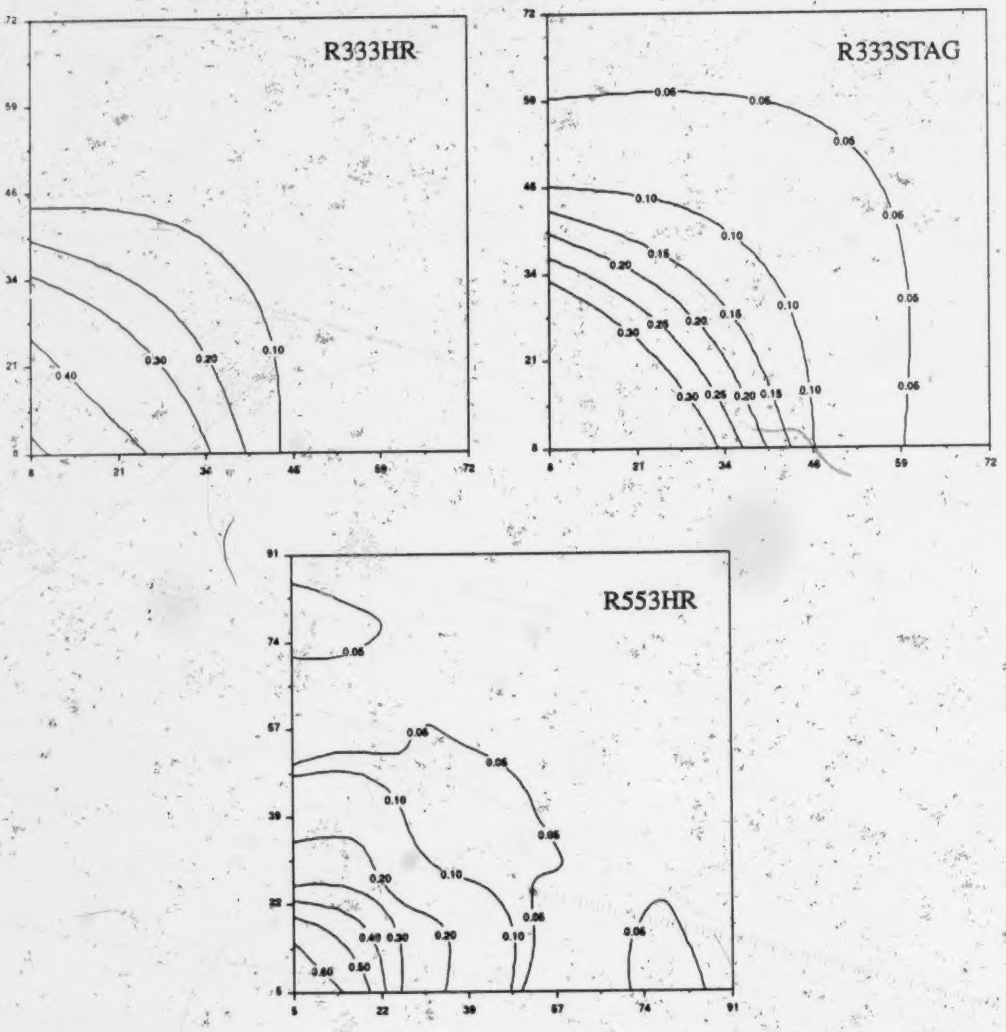


Figure 8–70

Contours of volumetric strain (%) in the high rate simulations from basal layer of Gauss points in heated zone.

the end of the second injection period in the high rate simulations are presented in Figure 8–70. The most striking item is that the contours for the uncoupled run R333HR and the staggered coupling run R333STAG are very similar. It would appear that running the coupled model has no advantage at this point, certainly if this were the only variable of interest. The volumetric strain contours in the fine grid simulations are much better delineated. While larger volumetric strains are noted at

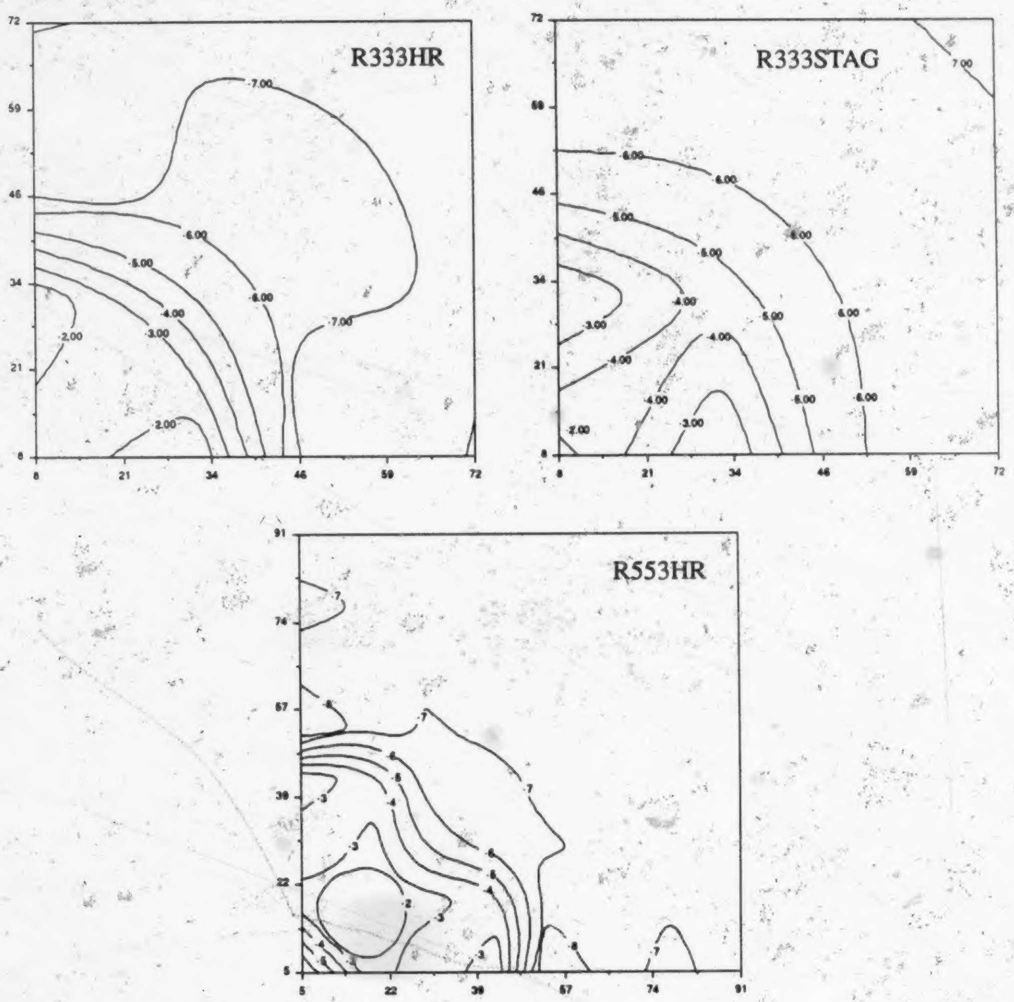


Figure 8–71

Contours of vertical stress (MPa) in the high rate simulations from basal layer of Gauss points in heated zone.

the wellbore, the values fall off with distance from the wellbore more quickly than for the coarser grid studies.

8.10.4 *Comparison of vertical stresses*

The vertical stresses, in conjunction with other information such as the three-dimensional stress ratio K_o, are important indicators of potential tensile hydraulic partings. As the injection of steam proceeds in a formation containing virtually immobile oil, the location and orientation of partings will govern the speed at which the oil is mobilized and produced. The distribution of vertical stresses in the bottom layer of Gauss points in the heated zone is presented for the three high-rate simulations in Figure 8–71. (Note that they are given in the tension-positive convention.) The distribution of vertical stress for the two coarse grid cases (one uncoupled, one coupled) is no longer nearly identical, as it was for the volumetric strain. The near wellbore stresses are higher for the coupled case, indicating that a coupled formulation will be necessary for a better prediction of where tensile partings might occur. The fine grid case gives a better resolution of the areal distribution of vertical stress. It can be seen that a low region (in magnitude) in the vertical stress occurs at some distance diagonally from the wellbore region, rather than at the wellbore. This has been noted before at later times in the coarse grid simulations by Tortike and Farouq Ali (1991).

8.11 *Making the coupled runs*

The coupled formulations presented in Chapter VII include using a pore compressibility term to enhance the implicitness of the fluid model solution. This approach causes fast divergence in the fully coupled case, although it was more stable for the staggered solution method. Many coupled runs were attempted, and most of them would diverge after only a few hours or days of simulation time. In the end, the coarse nature of the mesh required that multipliers be used to increase the volumetric strain used to calculate the porosity and permeabilities. The multiplier was used to convert the pore compressibility term from that back calculated from the solid material response in the uncoupled runs (ca. 3×10^{-6} psi^{-1}) to that value used in the uncoupled formulations (0.0001 psi^{-1}) — a factor of about 30. This made the comparisons easier as the initial pressure response was closer to the original uncoupled response, so that changes as a result of the coupling were more apparent.

The fully coupled run could not be made without adjusting it so that the solid velocity components remained constant at the values found after the first one or two Newton-Raphson iterations. This stabilized the solution considerably. Taking out what are implicit porosity terms in classical reservoir simulation to put them into the residual (the right-hand side of the equations) also makes the solution less stable, but was not such a big problem as the changes were less dramatic than in the solid velocities. The pore compressibility approach, which attempts to restore the implicit porosity terms using using the volumetric strains from the previous time step, failed to work at all in the fully coupled case. This is because the pore volume change is no longer only a function of the pore pressure in the block (and, implicitly, the neighbouring blocks), but is a function of the distribution of stresses throughout the entire body, which are transmitted at the speed of sound: hence the stress response to a pore pressure change cannot be considered a transient phenomenon. Therefore a change in stress at one corner of the body has an immediate effect on a position at the other corner (a finite distance away). Induced loads can then lead to changes in volumetric strain opposite to that deduced from the local change in pore pressure. Thus the implicit formulation leads to projections of porosity (a consequence of the Newton-Raphson formulation) which are more in error than allowing the porosity to lag one iteration (by placing the porosity terms into the residual).

It should also be noted that the staggered run should give the same answers as the fully coupled run if the time steps are short enough. At some point while reducing the time step size, the fully-coupled method will find a solution effectively within one Newton-Raphson iteration of the fluid flow model. When this occurs, the method is identical to the staggered scheme.

8.12 *Stress reduction calculations*

The reduction of stress to the yield surface is required because, by definition, the material cannot exist at stress states outside of the yield surface. The model finds stress states satisfying this condition by adjusting excess trial stresses, found by elastic projections) everywhere until the constraint is met. The approach used by Owen and Hinton (1982) is effective but can be slow. Typical mesh requirements are that the plastic zone be a small portion of the entire grid (< 10%–15%), and that loads be applied in small steps (e.g, < 5% of the total applied load). This allows the stresses to equilibrate throughout the entire body, through a redistribution of the excess stress in elastic space. The elastic space must absorb the excess energy existing as a result of reducing the stresses to the yield surface. If the stresses are not brought back to points on or within the yield surface, excess stresses are observed in the body with unreasonable strains and displacements in the same region. A better choice of stress reduction algorithm is then required, or lots of time or computing power.

8.13 *Stress sampling points*

Interpolation of stresses to the nodes is usually safe to do, although the values are not considered to be as accurate as at the optimal sampling points at the Gauss quadrature locations. Large errors can and do occur on the boundaries of the model, where the approximation procedure allows locally large gradients to give an unreasonably high answer. Values on the boundary should be compared to neighbouring Gauss points, and perhaps assigned values from these Gauss points.

8.14 *Usefulness of the runs less than fully coupled*

The uncoupled runs have been more useful than simply for base case studies. The stress and strain responses in the results presented here have been well within the same orders of magnitude observed for the coupled cases. As noticed earlier, there will be differences in the flow rates because of the modification of permeability with volumetric strain. It is also seen that the vertical stress distributions in the first two periods do not vary much. However, there is a significant difference between the stress ratio values. More differences do occur farther away from the wellbore, as can be seen in the contours discussed earlier. The uncoupled runs will continue to be useful to gauge the order of magnitude of the material response, which can be a guide to preparing the coupled model. This study indicates that the level of coupling achieved with the staggered formulation is good, and is considerably less costly to run than the fully coupled case.

8.15 *Computer time required to make runs*

The computer time required to make these runs has been described as excessive. The coupled models cannot be considered production versions. However, certain computational bottlenecks have been identified and can be modified to permit the completion of coupled runs in several hours or less, instead of several days. The excessive run times can be attributed entirely to the calculations in the geomechanical model.

First, the frontal solution method used by the program is not efficient for current computing facilities. The frontal solver assembles the global equations by processing the elements in a predetermined order. As global equations become complete, the steps for forward elimination are performed, so that it is complete by the time the the equations are prepared for the last element. Backward substitution for the solutions to the variables then proceeds as is normal in linear equation solvers. The combined process of global equation assembly and forward elimination, while effective in minimizing core storage, combines inefficient use of computer floating point arithmetic and a sole choice of linear equation solver — Gaussian elimination. Working with short vectors in FORTRAN `DO` loops is inefficient on modern workstations, and the generally piecemeal approach of the frontal solver consists entirely of this kind of operation. The frontal solver should be replaced by a two-step process of the complete assembly of the global stiffness matrix followed by the invocation of a suitable linear equation solver. The equation solver might be an iterative method with the appropriate choice of preconditioning and acceleration.

The second significant computational bottleneck is the explicit evaluation of stress for the stiffness matrix. This causes a large increase in the number of iterations required to reduce the excess trial stress (found by calculating the elastic stress response) to the yield surface. An implicit treatment of the stresses would enable a converged solution to be found more quickly, reducing the amount of computer time required.

Both these improvements can be implemented given time and effort. It was not done for this work because of limitations of time.

8.16 *Assumptions and limitations inherent in the models*

All the models described in this work contain implied as well as explicitly stated assumptions, which place bounds on the applicability of the models. In most cases, explicitly stated assumptions have been justified in the text, and the implication of the limitation discussed. In this section some of the wider implications are discussed, with some review, where important, of previously mentioned assumptions.

Obtaining material parameters is at least as difficult as measuring multiphase flow properties. It goes without saying that a lack of appropriate data values will alter the outcome of the simulation.

8.16.1 *Fluid and heat flow model*

The biggest limitation of the fluid flow model is that it is a dead oil model. The effect of gas evolution in the reservoir will be significant in terms of overall stiffness of the sand. It is also significant because it may create a mobile gas saturation through which steam can penetrate. These effects will not be modelled by the current simulator described in this work.

8.16.2 *Geomechanical model*

The principal limitation of the geomechanical model lies in the assumption of engineering (small) strain. Small strain analysis uses the undeformed length of the body to normalize the measure of strain. As a result, for strains greater than about 5%, there is an increasing error in the strain solution.

The use of constant elastic parameters limits the applicability of the geomechanical model. No scheme is incorporated into the model to account for stress dependent values of E or ν. However, hardening behaviour of the yield surface can be included.

Gridding a domain for plastic deformation analysis requires that the mesh be sufficiently large to redistribute and accommodate the excess strain energy transferred by elements with stress states on the yield point. A rule of thumb suggests that the mesh be sufficiently large so that no more than 10% to 15% of the elements are in the plastic region during the analysis.

The current geomechanical model cannot incorporate planes of weakness without using actual finite elements. In many formations shale streaks may be observed, which could cause displacements beyond the capabilities of the model. There is a general inadequacy in stress-strain constitutive modelling of soils when dealing with material which has been has been subjected to several load paths due to the rotation of principal stresses. The elastic and plastic parameters can be considered to have changed once principal stress rotation has occurred and the material is unloaded.

8.17 *Validation of the models*

The individual components of the finite difference/finite element coupled model were tested in their own right. The finite element model was tested using an axial loading example from Smith and Griffiths (1988), and was described earlier. The finite difference model was a production model used by Dr. Farouq Ali. After all the changes were made to the finite difference model, the program was tested by repeating a cyclic steam stimulation study. All the tests reproduced the original results exactly. This level of verification was considered adequate for the study, given the stated goals. Nevertheless, it is necessary to consider ways in which this complex model can be verified for production use. Field and laboratory conditions can be considered.

8.17.1 *Laboratory verification*

Two types of physical modelling could be prepared for testing the coupled numerical model. Standard drained and undrained tests in triaxial apparatus should be reproducible. Investigation of more complex processes would require appropriately designed models which would be able to display thermal and pore pressure gradients for the duration of the test. A two-dimensional model of one-quarter of the area covered by one steam injection well (injection in one corner of a square) could be designed which included strain gauges in the sand, along with pressure and temperature transducers.

8.17.2 *Verification using field data*

Full scale verification of this model is probably impossible at the moment. History matching single wellbore pressures, temperatures, and bitumen and water flow rates, does not guarantee a unique result. However, the model will ensure that the extent and degree of formation disturbance is quantified so that this information can be included in further reservoir modelling studies. Limited verification can be made by observing and matching surface uplift or compaction surrounding the wellbore, as well as comparing the extent of yielded material (from shear failure) to observations in surrounding wellbores, be they observation, injection or production wells. Stress and strain information from the wellbore cannot be included because of the local response by the formation to the presence of the wellbore and the fluid pressures in it, not accounted for in this model.

Chapter IX

CONCLUSIONS

Several numerical approaches for coupling oil reservoir fluid response and elastoplastic formation deformation have been considered. The first approach, a fully finite element formulation, was found to be too unstable to be useful. The second approach, a combination of finite element modelling of solid behaviour and finite difference modelling of thermal fluid flow, was found to be successful and useful. In particular, the following contributions can be identified and conclusions drawn:

i) this study has achieved the goal set out in Chapter III to develop a three-dimensional thermal numerical reservoir simulator incorporating the concomitant changes in stress and strain of an elastoplastic soil to hot fluid injection;

ii) this is the first implementation of non-linear, plastic behaviour in multiphase reservoir simulation using rigorous concepts of incremental plasticity;

iii) this is the first implementation of a three-dimensional multiphase reservoir model to observe the likely effects of principal stress rotation on potential parting planes;

iv) this study provides methods for quick appraisal and interpretation of three-dimensional stress results using simple two-dimensional plots;

v) the study demonstrates the rotation of principal stresses at orientations other than at the horizontal or vertical.

There are many recommendations for further research in this area. The more important ones are:

i) to find an implicit coupling term suitable for plastic formulations which incorporates the effects of stress changes;

ii) to investigate the effects of the presence of a solution gas in the oil;

iii) to implement a faster excess trial stress reduction algorithm;

iv) to increase the order of element for the finite element model;

v) to apply the model to existing field data, and to conduct laboratory tests which can be used for verification;

vi) to implement fracture-matrix interaction beyond the prediction of fracture initiation.

REFERENCES

Abou-Kassem, J.H. (1981): *Investigation of Grid Orientation in a Two-Dimensional, Compositional, Three-Phase Steam Model*, Ph.D. dissertation, Department of Chemical Engineering, University of Calgary.

Abou-Kassem, J.H., Farouq Ali, S.M., and Ferrer, J.C. (1986): "Appraisal of Steamflood Models", *Rev. Tec. Ing.* Univ. Zulia, Edicion Especial, **9** (2), 45–58.

Abramowitz, M., and Stegun, I.A. (1972): *Handbook of Mathematical Functions*, Dover Publications, Inc.

Agar, J.R. (1984): *Geotechnical Behaviour of Oil Sands at Elevated Temperatures and Pressures*, Ph.D. dissertation, Department of Civil Engineering, University of Alberta.

Agar, J.R., Morgenstern, N.R., and Scott, J.D. (1987): "Shear strength and stress-strain behaviour of Athabasca oil sand at elevated temperatures and pressures", *Can. Geotech. J.*, **24** 1–10.

Aktan, T. and Farouq Ali, S.M. (1978): "Finite Element Analysis of Temperature and Thermal Stresses Induced by Hot Water Injection", *SPEJ*, (December) 457–469.

Allen, M.B. (1984a): "Why Upwinding is Reasonable", 5th International Conference on Finite Elements in Water Resources, U of Vermont, (June) 13–23.

Allen, M.B. (1984b): *Collocation Techniques for Modeling Compositional Flows in Oil Reservoirs*, Springer-Verlag.

Allen, M.B. and Pinder, G.F. (1983): "Collocation Simulation of Multiphase Porous-Medium Flow", *SPEJ*, (February) 135–142.

Amyx, J.W., Bass, D.M., and Whiting, R.L. (1960): *Petroleum Reservoir Engineering*, McGraw-Hill, 99.

Annigeri, B.S., Cassenti, B.N., and Dennis, A.J. (1985): "Kinematics of Small and Large Deformations of Continua", *Eng. Comput.* **2** (September) 247–256.

Aziz, K., and Settari, A. (1973): *Petroleum Reservoir Simulation*, Applied Science Publishers Ltd.

Beattie, C.I., Boberg, T.C., and McNab, G.S. (1991): "Reservoir Simulation of Cyclic Steam Stimulation in the Cold Lake Oil Sands", *SPERE*, 6 2 (May) 200–206.

Biot, M.A. (1941): " General theory of three-dimensional consolidation", *J. Appl. Phys.* 12, 155–164.

Bird, R.B., Stewart, W.E., and Lightfoot, E.N. (1960): *Transport Phenomena*, J. Wiley and Sons, 313.

Brady, B.H.G, and Brown, E.T. (1985): *Rock Mechanics for Underground Mining*, Gearge Allen and Unwin, London.

Bronstein, I.N., and Semendyayev, K.A. (1985): *Handbook of Mathematics*, Verlag Harri Deutsch.

Burgess, K.A. (1978): "An Analysis of Semi-Direct Iterative and Direct Solution Techniques—An Application to the Cyclic Steam Stimulation Process", Ph.D. dissertation, Pennsylvania State University.

Cavendish, J.C., Price, H.S., and Varga, R.S. (1969): "Galerkin Methods for the Numerical Solution of Boundary Value Problems", *SPEJ* (June) 204–220.

Chalaturnyk, R. (1990): personal communication, Department of Civil Engineering, University of Alberta, Edmonton.

Chavant, G., Cohen, G., Jaffre, J., Dupuy, M. and Ribera, I. (1984): "Simulation of Two-Dimensional Waterflooding by Using Mixed Finite Elements", *SPEJ*, (August) 382–390.

Closmann, P.J., and Phocas, D.M. (1978): "Thermal Stresses Near a Heated Fracture in Transversely Isotropic Oil Shale", *SPEJ*, (February) 59–74.

Cohen, M.F. (1985): "Finite Element Methods for Enhanced Oil Recovery Simulation", SPE 13512, 8th SPE Symposium on Reservoir Simulation, Dallas, Texas (February).

Cook, R.D. (1981): *Concepts and applications of finite element analysis,*, 2nd. edition, J. Wiley & Sons, Inc.

Cooper, J.R., and Le Fevre, E.J. (1975): *Thermophysical Properties of Water Substance*, Edward Arnold (Publishers) Ltd, London.

Cowper, G.R.(1973): "Gaussian Quadrature Formulas for Triangles", *IJNME*, 7, 405–408.

Cuthill, E., and McKee, J. (1969): "Reducing the Bandwidth of Sparse Symmetric Matrices", Proceedings of 24th National Conference ACM, 157–172.

Dalen, V. (1979): "Simplified Finite-Element Models for Reservoir Flow Problems", *SPEJ* (October) 333–343.

Denbina, E.S., Boberg, T.C., and Rotter, M.B. (1991): "Evaluation of Key Reservoir Drive Mechanisms in the Early Cycles of Steam Stimulation at Cold Lake", *SPERE*, 6 2 (May) 207–211.

Dubois, P.F. (1982): "Swimming Upstream: Calculating Table Lookups and Piecewise Functions", in *Parallel Computations,* Ed. G. Rodrigue, Academic Press, 129–151.

Dusseault, M.B. (1977a): *The Geotechnical Characteristics of the Athabasca Oil Sands*, Ph.D. dissertation, Department of Civil Engineering, University of Alberta.

Dusseault, M. B. (1977b): "Stress State and Hydraulic Fracturing in the Athabasca Oil Sands", *JCPT* (July-Sept) 19–27.

Dusseault, M.B., and Morgenstern, N.R. (1978): "Shear Strength of Athabasca Oil Sands", *Can. Geot. J.*, 15 216–238.

Dusseault, M.B., and Simmons, J.V. (1982): "Injection-induced stress and fracture orientation changes", *Can. Geotech. J.*, 19 (May) 483–493.

Dusseault, M.B., and Rothenburg, L. (1988): "Shear Dilatancy and Permeability Enhancement in Oil Sands", Paper no. 32 in Fourth UNITAR/UNDP International Conference on Heavy Crude and Tar Sands (7–12 August 1988).

Everstine, G.C. (1979): "A Comparison of Three Resequencing Algorithms for the Reduction of Matrix Profile and Wavefront", *IJNME* 14 837–853.

Ewing, R.E. (1989): "Finite Elements in Reservoir Simulation," in Finite Element Analysis in Fluids, Proc. of the 7th International Conference on Finite Element Methods in Flow Problems, U. of Alabama, Huntsville, Alabama. 1251–1257.

Faust, C.R. and Mercer, J.W. (1976): "An Analysis of Finite-Difference and Finite-Element Techniques for Geothermal Reservoir Simulation", SPE 5742 (February).

Farouq Ali, S.M. (1982): *Elements of Heavy Oil Recovery,* Department of Mineral Engineering, University of Alberta, Canada.

Feda, J. (1982): *Mechanics of Particulate Materials,* Elsevier, The Netherlands, p267.

Finol, A., and Farouq Ali, S.M. (1975): "Numerical Simulation of Oil Production with Simultaneous Ground Subsidence", *SPEJ,* 411 (October).

Ferrer, J., Quintero, L., Aquino, C., Guevara, F., Avila, A., Pérez, J., Briceño, W., Sánchez, I., Rodriquez, E., Finol, A., Diaz-Muñoz, J., and Farouq Ali, S.M. (1980): "Development and Application of Steam Injection Simulators in Venezuela", CIM paper no. 80-31-09 presented at 31st Ann. Tech. Mtg. Pet. Soc. CIM, Calgary, AB, 25–28 May 1980.

Frederick, D., and Chang, T.S. (1965): *Continuum Mechanics,* Allyn and Bacon.

Fung, L.S.K. (1990): "A Coupled Gemechnaical Multiphase FLow Model for Analysis of In-Situ Recover in Cohesionless Oil Sands", CIM/SPE 90-29 presented at the CIM/SPE 1990 International Technical Meeting, Calgary, Alberta (10–13 June 1990).

Gassman, F. (1951): "Über die Elastizität poröser Medien", Vierteljahrsschrift der Naturforschenden Gesellshaft in Zürich **96** 1; *Mitteilungen aus dem Institut für Geophysik* **17** 1.

Geertsma, J. (1957): "The Effect of Fluid Pressure Decline on Volumetric Changes of Porous Rocks", *Pet. Trans. AIME,* **210** 331–340.

Geertsma, J. (1966): "Problems of rock mechanics in petroleum production engineering", in Proceedings of the 1st Congress of the International Society of Rock Mechanics, Lisbon, 585–594.

Gentzsch, W. (1984): *Vectorization of Computer Programs with Applications to Computational Fluid Mechanics,* Friedr. Vieweg & Sohn, Braunschweig/Wiesbaden, 81.

Gibbs, N.E. (1976): "ALGORITHM 509 A Hybrid Profile Reduction Algorithm [FI]", *ACM Transactions on Mathematical Software* **2** (4) (December) 378–387.

Gibbs, N.E., Poole, W.G., and Stockmeyer, P.K. (1976): "An Algorithm for Reducing the Bandwidth and Profile of a Sparse Matrix", *SIAM J Numerical Analysis* **13** (2) (April) 236–250.

Gottfried, B.S. (1965): "A Mathematical Model of Thermal Oil Recovery in Linear Systems", *SPEJ* (September) 196–210.

Gray, W.G. (1975): "A Derivation of the Equations for Multi-Phase Transport", *Chemical Engineering Science*, **30** 229–233.

Gray, W.G., and O'Neill, K. (1976): "On the General Equations for Flow in Porous Media and Their Reduction to Darcy's Law", *Water Resources Research*, **12** (2) (April) 148–154.

Gulbrandsen, S., and Wille, S.Ø. (1985): "A Finite Element Formulation of the Two-Phase Flow Equations for Oil Reservoirs", SPE 13516, 8th SPE Symposium on Reservoir Simulation, Dallas, Texas (February).

Hayes, L.J., Kendall, R.P. and Wheeler, M.F. (1977): "The Treatment of Sources and Sinks in Steady-State Reservoir Engineering Simulations", Advances in Computer Methods for Partial Differential Equations III, Ed by R. Vichnevetsky, IMACS (AICA).

Hinton, E., and Campbell, J.S. (1974): "Local and Global Smoothing of Discontinuous Finite Element Functions Using a Least Squares Method", *IJNME* **8** 461–480.

Hinton, E., and Owen, D.R.J. (1977): *Finite Element Programming*, Academic Press, London.

Hill, R. (1950): *The Mathematical Theory of Plasticity*, Oxford University Press, Oxford.

Hoek, E., and Brown, E.T. (1980): *Underground Excavations in Rock*, Institution of Mining and Metallurgy, London.

Huyakorn, P.S. (1977): "Solution of steady-state, convective transport equation using an upwind finite element scheme", Applied Math. Modelling, 1, 187–195 (March).

Huyakorn, P.S., and Nilkuha, K. (1979): "Solution of transient transport equation using an upstream finite element scheme", Applied Math. Modelling, 3, 7–17 (February).

Huyakorn, P.S., and Pinder, G.F. (1983): *Computational Methods in Subsurface Flow*, Academic Press, 229–254.

Huyakorn, P.S. and Pinder, G.F. (1977a): "Solution of Two-Phase Flow Using a New Finite Element Technique", Applied Numerical Modelling, Ed. by C. A. Brebbia, U. of Southampton, (July) 375–390.

Huyakorn, P.S. and Pinder, G.F.(1977b): "A Pressure-Enthalpy Finite Element Model for Simulating Hydrothermal Reservoirs", Advances in Computer Methods for Partial Differential Equations II, R. Vichnevetsky (Ed), IMACS (AICA).

Huyakorn, P.S. and Pinder, G.F. (1978): "A New Finite Element Technique for the Solution of Two-Phase Flow through Porous Media", *Advances in Water Resources*, 1 (5) 285–298.

Irons, B.M. (1969): "Economical Computer Techniques for Numerically Integrated Finite Elements", IJNME, 1, 201–203.

Ito, Y. (1984): "The Introduction of the Microchanneling Phenomenon to Cyclic Steam Stimulation and Its Application to the Numerical Simulator (Sand Deformation Concept)", *SPEJ*, 417–429 (August).

Javandel, I., and Witherspoon, P.A. (1968): "Application of the Finite Element Method to Transient Flow in Porous Media", SPEJ, 241–252 (September).

Karahanoglu, N., Doyuran, V. and Akkas, N. (1985): "Finite Element Simulation of Hot-Water-Type Geothermal Reservoirs", *J. Volcanology and Geothermal Research*, 23 357–382.

Knuth, D.E. (1981): *The Art of Computer Programming. Volume 2 / Seminumerical Algorithms*, 2nd ed., Addison-Wesley Publishing Company, Reading, Massachussetts, 647.

Kosar, K.M. (1984): *Geotechnical Behaviour of Oil Sands at Elevated Temperatures and Pressures*, Ph.D. dissertation, Department of Civil Engineering, University of Alberta.

Kosar, K.M., Scott, J.D., and Morgenstern, N. R. (1987): "Testing to Determine the Geotechnical Properties of Oil Sands", CIM Paper 87-38-59, 38th ATM of the Pet. Soc. CIM, Calgary (7–10 June).

Kreysig, E. (1988): *Advanced Engineering Mathematics*, 6th edition, John Wiley & Sons, New York.

Lade, P.V., and Nelson, R.B. (1984): "Incrementalization Procedure for Elasto-Plastic Constitutive Model with Multiple, Intersecting Yield Surfaces", *Int. J. Num. Anal. Meth. Geomech.*, 8 311–323.

Lewis, J.G. (1982a): "ALGORITHM 582 The Gibbs-Poole-Stockmeyer and Gibbs-King Algorithms for Reordering Sparse Matrices", *ACM Transactions on Mathematical Software*, 8 (2) (June) 190–194.

Lewis, J.G. (1982b): "Implementation of the Gibbs-Poole-Stockmeyer and Gibbs-King Algorithms", *ACM Transactions on Mathematical Software*, **8** (2) (June) 180–189.

Lewis R.W., Morgan, K. and Roberts, P.M. (1985): "Finite Element Simulation of Thermal Recovery Processes and Heat Losses to Surrounding Strata", 3rd European Meeting on Improved Oil Recovery, Rome, (April) 305–315.

Lewis R.W., White, I.R. and Wood, W.L. (1978): "A Starting Algorithm for the Numerical Simulation of Two-Phase Flow Problems", *IJNME* **12** 319–328.

Lode, W. (1926): "Versuche über den Einfluß der mittleren Hauptspannung auf das Fließ in der Metalle Eisen, Kupfer und Nickel", Zeitschrift für Physik **36** 913–939.

Logie, R.B., Ranganathan, R., Pruden, B.B. and Denis, J.M. (1978): "Comparison of Thermal Hydrocracking with Thermal Cracking of Athabasca Bitumen at Low Conversions", CANMET report 78-20, AOSIC# 006626 (October).

Mase, G.E. (1970): *Schaum's Outline of Theory and Problems of Continuum Mechanics*, McGraw-Hill Book Co., New York.

McMichael, C.L. and Thomas, G.W. (1973): "Reservoir Simulation by Galerkin's Method", *SPEJ* (June) 125–138.

Mohsen, M.F.N. and Pinder, G.F. (1983): "Collocation with 'Adaptive' Finite Elements in Multi-Phase Porous Medium Flow", SPE 12345 (June).

Mori, A., and Tamura, M. (1986): "Effect of dilatancy on permeability in sands stabilized by chemical grout", *Soils and Foundations*, Japanese Soc. of Soil Mechanics and Foundation Engg, **26** 1, 96–104.

Morris, A H. (1990): NSWC Library of Mathematics Subroutines (Public release no. NSWC TR 90-21), Naval Surface Warfare Center, Dahlgren, Virginia 22448-5000, U.S.A.

Nash, W.A. (1977): *Schaum's Outline of Theory and Problems of Strength of Materials*, 2nd edition, McGraw-Hill Book Co., New York.

Naylor, D.J. (1978): "Stress-Strain Laws for Soil", in C. R. Scott (Eds.), *Developments in Soil Mechanics — 1*, Applied Science Publishers, London, 39–68.

Nolen, J.S., and Berry, D.W.: "Tests of the Stability and Time-Step Sensitivity of Semi-Implicit Reservoir Simulation Techniques", *SPEJ* (June) 253–266.

Owen, D.R.J., and Hinton, E. (1982): *Finite Elements in Plasticity*, Pineridge Press, Swansea.

Perry, R.H., and Green, D. (1984): *Perry's Chemical Engineers' Handbook*, 6th edition, McGraw-Hill Book Co., New York.

Pietruszczak, S., and Krucinski, S. (1989): "Considerations on Soil Response to the Rotation of Principal Stress Directions", *Computers and Geotechnics*, 8 89–110.

Prats, M. (1982): *Thermal Recovery*, SPE Monograph No 7, Dallas.

Price, H.S., and Coats, K.H. (1974): "Direct Methods in Reservoir Simulation", SPEJ (June) 295–308.

Pyrah, I.C. (1987): "Elasto-Plastic Analysis in Geotechnical Engineering using Finite Differences", Computational Plasticity: Models, Software and Applications, edited by D.R.J. Owen, E. Hinton, and E. Onate, 1607–1620.

Raghavan, R. (1972): "A Review of the Consolidation and Rebound Processes in One-Dimensional Porous Columns", SPE 4078 presented at the 47th Annual Fall Meeting of the SPE, San Antonio, Texas.

Rattia-Regalado, A.J. (1980): *Numerical Simulation of Steam Injection into a Compacting Reservoir under Non-Newtonian Flow Conditions*, Ph.D. Dissertation, Pennsylvania State University.

Reid, R.C., Prausnitz J.M., and Sherwood, T.K. (1977): *The Properties of Gases and Liquids*, 3rd edition, McGraw-Hill Book Co.

Rice, J.R. (1983): *Numerical Methods, Software, and Analysis*, McGraw-Hill Book Co., 167.

Schrefler, B.A., and Simoni, L. (1988): "A Unified Approach to the Analysis of Saturated-Unsaturated Elastoplastic Porous Media", in *Numerical methods in Geomechanics (Innsbruck 1988)*,, G. Swoboda (Ed.), I 205–212.

Schrefler, B.A., and Simoni, L. (1991): "Comparison between different finite element solutions for immiscible two-phase flow in deforming porous media", in *Computer Methods and Advances in Geomechanics (Cairns 1991)*, G. Beer, J.R. Booker, J.P. Carter (Eds.), **2** 1215–1220.

Scott, J.D., Adhikary, D., and Proskin, S.A. (1991): "Volume and Permeability Changes Associated with Steam Stimulation in an Oil Sands Reservoir", paper CIM/AOSTRA 91-63 presented at the CIM/AOSTRA 1991 Technical Conference, Banff, Alberta (21–24 April 1991).

Settari, A. (1988): "Modelling of Fracture and Deformation Processes in Oil Sands", Paper no. 43 in Fourth UNITAR/UNDP International Conference on Heavy Crude and Tar Sands (7–12 August 1988).

Settari, A., Kry, P.R., and Yee, C.-T. (1989): " Coupling of Fluid Flow and Soil Behaviour to Model Injection into Uncemented Oil Sands", *J. Can. Pet. Tech.*,, **28** 1 (Jan.–Feb.) 81–92.

Settari, A., (1989): "Physics and Modeling of Thermal Flow and Soil Mechanics in Unconsolidated Porous Media", SPE 18420 presented at the Tenth SPE Symposium on Reservoir Simulation, Houston, TX (6–8 February 1989).

Slattery, J.C. (1972): *Momentum, Energy and Mass Transfer in Continua*, McGraw-Hill Chemical Engineering Series.

Smith, I.M., and Griffiths, D.V. (1988): *Programming the Finite Element Method*, 2nd edition, J. WIley & Sons, Chichester.

Smith-Magowan, D., Skauge, A., and Hepler, L. (1982): "Specific heats of Athabasca oil sands and components", *JCPT* (May-June) 28–32.

Spivak, A., Price, H.S. and Settari, A. (1977): "Solution of the Equations for Multidimensional, Two-Phase, Immiscible Flow by Variational Methods", *SPEJ* (February) 27–41.

Strang, G., and Fix, G.J. (1973): *An analysis of the finite element method*, Prentice-Hall, Englewood Cliffs, N.J.

Svrcek, W.Y., and Mehrotra, A.K. (1982): "Gas solubility, viscosity and density measurements for Athabasca bitumen", *JCPT* (July–August) 31–38.

Timoshenko, S.P., and Goodier, J.N. (1982): *Theory of Elasticity,* McGraw-Hill Book Co., New York.

Tortike, W.S. (1984): *Post-Communication Development Recovery Processes at GLISP—a Numerical Study,* M.Eng. dissertation, University of Alberta, 85–89.

Tortike, W.S., and Farouq Ali, S.M. (1987): "STAIRWAY: an effective sparse Gaussian solver for the Cyber 205 vector computer", presented at *iciam '87,* the first international conference on industrial and applied mathematics, Paris, France (29 June–3 July).

Tortike, W. S., and Farouq Ali, S. M. (1989): "Saturated Steam Property Functional Correlations for Fully Implicit Thermal Reservoir Simulation", SPE Reservoir Engineering, **4** 4 471–474 (November 1989).

Touloukian, Y.S., Kirby, R.K., Taylor, R.E. and Lee, T.Y.R. (1970): *Thermophysical Properties of Matter,* Vols. 1–11, IFI/Plenum.

Vaziri, H.H. (1988): "Coupled Fluid Flow and Stress Analysis of Oil Sands Subject to Heating", *J. Can. Pet. Tech.,* **27** 5 (Sept.–Oct.) 84–91.

Vermeer, P.A., and de Borst, R. (1984): "Non-associated plasticity for soils, concrete and rock", *HERON,* **29** 3, 1–64.

Vesic, A.S., and Clough, G.W. (1968): "Behavior of Granular Materials under High Stresses", *Journal of the Soil Mechanics and Foundations Division,* Proceedings of the American Society of Civil Engineers, **94** SM 3, 661–688.

Voss, C.I. (1978): "Finite Element Simulation of Multiphase Geothermal Reservoirs, Ph.D. dissertation, DAI #7823526, Department of Civil Engineering, Princeton University, New Jersey.

Wan, R., Chan, D.H., and Kosar, K.M. (1989): "A Constitutive Model for the Effective Stress-Strain Behaviour of Oil Sands", Paper no. 89-40-66, 40th ATM Pet. Soc. of CIM, Banff (28–31 May).

Westlake, J.R. (1968): *A Handbook of Numerical Matrix Inversion and Solution of Linear Equations,* John Wiley and Sons.

Whitaker, S. (1973): "The Transport Equations for Multiphase Systems", *Chemical Engineering Science* 28 139–147.

White, I.R., Lewis, R.W. and Wood, W.L. (1981): "The numerical simulation of multiphase flow through a porous medium and its application to reservoir engineering", *Applied Mathematical Modelling*, 5 (June) 165–172.

Wilson, D.C. and Casinder, P.C. (1978): "The Use of Finite Element Models in Simulating North Sea Waterfloods", SPE 8111, Europec, London.

Wong, T.T. (1989): *personal communication*, Department of Civil Engineering, University of Alberta, Edmonton.

Wood, D.M. (1990): *Soil behaviour and critical state soil mechanics*, Cambridge University Press, Cambridge.

Woodland, D.C., and Bell, J.S. (1989): "In-Situ Stress Magnitudes from Mini-Frac Records in Western Canada", *JCPT* 28 5 (Sept.–Oct.) 22–31.

Xikui, L., Zienkiewicz, O.C., and Xie, Y.M. (1990): "A Numerical Model for Immiscible Two-Phase Fluid Flow in a Porous Medium and its Time Domain Solution", *IJNME* 30 1195–1212.

Zienkiewicz, O.C. (1977): *The Finite Element Method*, 3rd ed., McGraw-Hill, London, 27.

Zienkiewicz, O.C., and Morgan, K. (1983): *Finite Elements and Approximations'*, J. Wiley and Son, 38–49.

Zienkiewicz, O.C. and Parekh, C.J. (1970): "Transient Field Problems: Two Dimensional and Three Dimensional Analysis by Isoparametric Finite Elements", *IJNME* 2 61–71.

Appendix A

COUPLED FLOW AND SOLID EQUATIONS

The following set of flow equations are used in the finite element formulation. The primary variables (S_w, S_o, p_w, T) are solved for at the latest time level. The iterative solution allows the non-linear terms to be evaluated at the last iteration solved. Terms at the previous time level are denoted by the superscript $^\nu$, and are constants, not variables. See the Nomenclature and Chapter IV for further explanation of symbols used.

A.1 *Water equation*

The equation for water can be written as

$$A_1 \frac{\partial S_w}{\partial t} + A_2 \frac{\partial S_o}{\partial t} + A_3 \frac{\partial p_w}{\partial t} + A_4 \frac{\partial T}{\partial t} + A_5 S_w + A_6 S_o + \vec{\nabla} \cdot (\mathbf{A}_7 \cdot \vec{\nabla} p_w) + \vec{\nabla} \cdot \mathbf{A}_8 + A_9 + A_{10} \frac{\partial (\vec{\nabla} \cdot \mathbf{u})}{\partial t} = 0. \tag{A.1}$$

where

$$A_1 = \phi \rho_w - \phi y_{wv} \alpha_v \tag{A.2}$$

$$A_2 = -\phi y_{wv} \alpha_v \tag{A.3}$$

$$A_3 = \phi \left[\frac{\partial \rho_w}{\partial p_w} S_w^\nu + \frac{\partial (y_{wv} \alpha_v)}{\partial p_w} (1 - S_o^\nu - S_w^\nu) \right] \tag{A.4}$$

$$A_4 = \phi \frac{\partial \rho_w}{\partial T} S_w^\nu + \phi \frac{\partial (y_{wv} \alpha_v)}{\partial T} (1 - S_o^\nu - S_w^\nu) + [\rho_w S_w^\nu + y_{wv} \alpha_v (1 - S_o^\nu - S_w^\nu)] \frac{1 - \phi}{\rho_r} \frac{\partial \rho_r}{\partial T} \tag{A.5}$$

$$\begin{aligned} A_5 = \left(\frac{\partial \rho_w}{\partial p_w} - \frac{\partial (y_{wv} \alpha_v)}{\partial p_w} \right) \phi \frac{\partial p_w}{\partial t}^{\nu} + \left(\phi \frac{\partial \rho_w}{\partial T} - \phi \frac{\partial (y_{wv} \alpha_v)}{\partial T} + (\rho_w - y_{wv} \alpha_v) \frac{1 - \phi}{\rho_r} \frac{\partial \rho_r}{\partial T} \right) \frac{\partial T}{\partial t}^{\nu} \\ + (\rho_w - y_{wv} \alpha_v) \frac{\partial (\vec{\nabla} \cdot \mathbf{u})}{\partial t}^{\nu} \end{aligned} \tag{A.6}$$

$$A_6 = - \left[\phi \left(\frac{\partial (y_{wv} \alpha_v)}{\partial p_w} \frac{\partial p_w}{\partial t}^{\nu} + \frac{\partial (y_{wv} \alpha_v)}{\partial T} \frac{\partial T}{\partial t}^{\nu} \right) + y_{wv} \alpha_v \left(\frac{\partial (\vec{\nabla} \cdot \mathbf{u})}{\partial t}^{\nu} + \frac{1 - \phi}{\rho_r} \frac{\partial \rho_r}{\partial T} \frac{\partial T}{\partial t}^{\nu} \right) \right] \tag{A.7}$$

$$\mathbf{A}_7 = -\mathbf{k} (\rho_w \lambda_w + y_{wv} \alpha_v \lambda_v) \tag{A.8}$$

$$\mathbf{A}_8 = -\mathbf{k} (\rho_w^2 \lambda_w + y_{wv} \alpha_v \rho_v \lambda_v) \cdot \mathbf{g} \tag{A.9}$$

$$\begin{aligned} A_9 = \left[\phi \frac{\partial p_w}{\partial t}^{\nu} \frac{\partial (y_{wv} \alpha_v)}{\partial p_w} + \phi \frac{\partial T}{\partial t}^{\nu} \frac{\partial (y_{wv} \alpha_v)}{\partial T} + y_{wv} \alpha_v \left(\frac{\partial (\vec{\nabla} \cdot \mathbf{u})}{\partial t}^{\nu} + \frac{1 - \phi}{\rho_r} \frac{\partial \rho_r}{\partial T} \frac{\partial T}{\partial t}^{\nu} \right) \right] (S_w^\nu + S_o^\nu) \\ - S_w^\nu \left[\phi \left(\frac{\partial p_w}{\partial t}^{\nu} \frac{\partial \rho_w}{\partial p_w} + \frac{\partial T}{\partial t}^{\nu} \frac{\partial \rho_w}{\partial T} \right) + \rho_w \left(\frac{1 - \phi}{\rho_r} \frac{\partial \rho_r}{\partial T} \frac{\partial T}{\partial t}^{\nu} + \frac{\partial (\vec{\nabla} \cdot \mathbf{u})}{\partial t}^{\nu} \right) \right] \end{aligned} \tag{A.10}$$

$$A_{10} = \rho_w S_w^\nu + y_{wv} \alpha_v (1 - S_w^\nu - S_o^\nu) \tag{A.11}$$

A. 2 *Bitumen equation*

The equation for bitumen can be written as

$$B_1\frac{\partial S_o}{\partial t}+B_2\frac{\partial p_w}{\partial t}+B_3\frac{\partial T}{\partial t}+B_4S_o+\vec{\nabla}\cdot(\mathbf{B}_5\cdot\vec{\nabla}p_w)+\vec{\nabla}\cdot(\mathbf{B}_6:\vec{\nabla}S_o)+\vec{\nabla}\cdot\mathbf{B}_7+B_8+B_9\frac{\partial(\vec{\nabla}\cdot\mathbf{u})}{\partial t}=0. \tag{A.12}$$

where

$$B_1=\phi\rho_b \tag{A.13}$$

$$B_2=\phi S_o^\nu\frac{\partial\bar{\rho}_b}{\partial p_w} \tag{A.14}$$

$$B_3=\phi S_o^\nu\frac{\partial\bar{\rho}_b}{\partial T}+\bar{\rho}_b\frac{1-\phi}{\rho_r}\frac{\partial\rho_r}{\partial T}S_o^\nu \tag{A.15}$$

$$B_4=\phi\left(\frac{\partial p_w}{\partial t}^\nu\frac{\partial\bar{\rho}_b}{\partial p_w}+\frac{\partial T}{\partial t}^\nu\frac{\partial\bar{\rho}_b}{\partial T}\right)+\bar{\rho}_b\left(\frac{\partial(\vec{\nabla}\cdot\mathbf{u})}{\partial t}^\nu+\frac{1-\phi}{\rho_r}\frac{\partial\rho_r}{\partial T}\frac{\partial T}{\partial t}^\nu\right) \tag{A.16}$$

$$\mathbf{B}_5=-\lambda_o\hat{\rho}_b\mathbf{k} \tag{A.17}$$

$$\mathbf{B}_6=-\lambda_o\hat{\rho}_b\mathbf{k}\frac{\partial P_{c_{ow}}}{\partial S_o} \tag{A.18}$$

$$\mathbf{B}_7=-(\mathbf{k}\hat{\rho}_b\rho_o\lambda_o)\,\mathbf{g} \tag{A.19}$$

$$B_8=-S_o^\nu\left[\phi\left(\frac{\partial p_w}{\partial t}^\nu\frac{\partial\hat{\rho}_b}{\partial p_w}+\frac{\partial T}{\partial t}^\nu\frac{\partial\hat{\rho}_b}{\partial T}\right)+\hat{\rho}_b\left(\frac{\partial(\vec{\nabla}\cdot\mathbf{u})}{\partial t}^\nu+\frac{1-\phi}{\rho_r}\frac{\partial\rho_r}{\partial T}\frac{\partial T}{\partial t}^\nu\right)\right] \tag{A.20}$$

$$B_9=\hat{\rho}_bS_o^\nu \tag{A.21}$$

A. 3 *Gas equation*

The equation for gas can be written as

$$C_1\frac{\partial S_w}{\partial t}+C_2\frac{\partial S_o}{\partial t}+C_3\frac{\partial p_w}{\partial t}+C_4\frac{\partial T}{\partial t}+C_5S_w+C_6S_o+\vec{\nabla}\cdot(\mathbf{C}_7\cdot\vec{\nabla}p_w)+\vec{\nabla}\cdot(\mathbf{C}_8\cdot\vec{\nabla}S_o)$$
$$+\vec{\nabla}\cdot\mathbf{C}_9+C_{10}+C_{11}\frac{\partial(\vec{\nabla}\cdot\mathbf{u})}{\partial t}=0 \tag{A.22}$$

where

$$C_1=-\phi y_{gv}\beta_v \tag{A.23}$$

$$C_2=\phi(\hat{\rho}_g-y_{gv}\beta_v) \tag{A.24}$$

$$C_3=\phi\left[\frac{\partial\hat{\rho}_g}{\partial p_w}S_o^\nu+\frac{\partial(y_{gv}\beta_v)}{\partial p_w}(1-S_w^\nu-S_o^\nu)\right] \tag{A.25}$$

$$C_4=\phi\left[\frac{\partial\hat{\rho}_g}{\partial T}S_o^\nu+\frac{\partial(y_{gv}\beta_v)}{\partial T}(1-S_w^\nu-S_o^\nu)\right]+\left[\hat{\rho}_gS_o^\nu+y_{gv}\beta_v(1-S_w^\nu-S_o^\nu)\right]\frac{1-\phi}{\rho_r}\frac{\partial\rho_r}{\partial T} \tag{A.26}$$

$$C_5=-\left[\phi\left(\frac{\partial(y_{gv}\beta_v)}{\partial p_w}\frac{\partial p_w}{\partial t}^\nu+\frac{\partial(y_{gv}\beta_v)}{\partial T}\frac{\partial T}{\partial t}^\nu\right)+y_{gv}\beta_v\left(\frac{\partial(\vec{\nabla}\cdot\mathbf{u})}{\partial t}^\nu+\frac{1-\phi}{\rho_r}\frac{\partial\rho_r}{\partial T}\frac{\partial T}{\partial t}^\nu\right)\right] \tag{A.27}$$

$$C_6=\left(\frac{\partial\hat{\rho}_g}{\partial p_w}-\frac{\partial(y_{gv}\beta_v)}{\partial p_w}\right)\phi\frac{\partial p_w}{\partial t}^\nu+\left(\frac{\partial\hat{\rho}_g}{\partial T}-\frac{\partial(y_{gv}\beta_v)}{\partial T}\right)\phi\frac{\partial T}{\partial t}^\nu+$$
$$(\hat{\rho}_g-y_{gv}\beta_v)\left(\frac{\partial(\vec{\nabla}\cdot\mathbf{u})}{\partial t}^\nu+\frac{1-\phi}{\rho_r}\frac{\partial\rho_r}{\partial T}\frac{\partial T}{\partial t}^\nu\right)\Bigg] \tag{A.28}$$

$$\mathbf{C}_7=-\mathbf{k}(\hat{\rho}_g\lambda_o+y_{gv}\beta_v\lambda_v) \tag{A.29}$$

$$\mathbf{C}_8=-\mathbf{k}\hat{\rho}_g\lambda_o\frac{\partial P_{c_{ow}}}{\partial S_o} \tag{A.30}$$

$$\mathbf{C}_9=-\mathbf{k}(\hat{\rho}_g\lambda_o\rho_o+y_{gv}\beta_v\lambda_v\rho_v)\cdot\mathbf{g} \tag{A.31}$$

$$C_{10}=(S_w^\nu+S_o^\nu)\left[\phi\frac{\partial p_w}{\partial t}^\nu\frac{\partial(y_{gv}\beta_v)}{\partial p_w}+\phi\frac{\partial T}{\partial t}^\nu\frac{\partial(y_{gv}\beta_v)}{\partial T}+y_{gv}\beta_v\left(\frac{\partial(\vec{\nabla}\cdot\mathbf{u})}{\partial t}^\nu+\frac{1-\phi}{\rho_r}\frac{\partial\rho_r}{\partial T}\frac{\partial T}{\partial t}^\nu\right)\right]$$
$$-S_o^\nu\left[\phi\frac{\partial p_w}{\partial t}^\nu\frac{\partial\hat{\rho}_g}{\partial p_w}+\phi\frac{\partial T}{\partial t}^\nu\frac{\partial\hat{\rho}_g}{\partial T}+\hat{\rho}_g\left(\frac{\partial(\vec{\nabla}\cdot\mathbf{u})}{\partial t}^\nu+\frac{1-\phi}{\rho_r}\frac{\partial\rho_r}{\partial T}\frac{\partial T}{\partial t}^\nu\right)\right] \tag{A.32}$$

$$C_{11} = \hat{\rho}_g S_o^{\nu} + y_{gv} \beta_v (1 - S_w^{\nu} - S_o^{\nu}) \tag{A.33}$$

A. 4 *Energy equation*

The equation for energy can be written as

$$D_1 \frac{\partial S_o}{\partial t} + D_2 \frac{\partial p_w}{\partial t} + D_3 \frac{\partial T}{\partial t} + D_4 S_w + D_5 S_o + D_6 p_w + \mathbf{D}_7 \cdot \vec{\nabla} S_o + \mathbf{D}_8 \cdot \vec{\nabla} p_w + \mathbf{D}_9 \cdot \vec{\nabla} T$$
$$+ \vec{\nabla} \cdot (D_{10} \nabla T) + D_{11} \frac{\partial (\vec{\nabla} \cdot \mathbf{u})}{\partial t} + \mathbf{D}_{12} \cdot \frac{\partial \mathbf{u}}{\partial t} + D_{13} = 0, \quad \text{(A.34)}$$

where

$$D_1 = -\phi \left(P_{c_{ow}} + \frac{\partial P_{c_{ow}}}{\partial S_o} S_o^{\nu} \right) \quad \text{(A.35)}$$

$$D_2 = (1-\phi) \rho_r \frac{\partial h_r}{\partial p_w} - \phi + \phi \left(S_w^{\nu} \rho_w \frac{\partial h_w}{\partial p_w} + S_o^{\nu} \rho_o \frac{\partial h_o}{\partial p_w} + (1 - S_w^{\nu} - S_o^{\nu}) \rho_v \frac{\partial h_v}{\partial p_w} \right) \quad \text{(A.36)}$$

$$D_3 = (1-\phi) \rho_r \frac{\partial h_r}{\partial T} + \phi \left(S_w^{\nu} \rho_w \frac{\partial h_w}{\partial T} + S_o^{\nu} \rho_o \frac{\partial h_o}{\partial T} + (1 - S_w^{\nu} - S_o^{\nu}) \rho_v \frac{\partial h_v}{\partial T} \right)$$
$$+ \frac{1-\phi}{\rho_r} \frac{\partial \rho_r}{\partial T} (p_w^{\nu} + S_o^{\nu} P_{c_{ow}}) \quad \text{(A.37)}$$

$$D_4 = \phi \left[\frac{\partial p_w}{\partial t}^{\nu} \left(\rho_w \frac{\partial h_w}{\partial p_w} - \rho_v \frac{\partial h_v}{\partial p_w} \right) + \frac{\partial T}{\partial t}^{\nu} \left(\rho_w \frac{\partial h_w}{\partial T} - \rho_v \frac{\partial h_v}{\partial T} \right) \right] \quad \text{(A.38)}$$

$$D_5 = \phi \left[\frac{\partial p_w}{\partial t}^{\nu} \left(\rho_o \frac{\partial h_o}{\partial p_w} - \rho_v \frac{\partial h_v}{\partial p_w} \right) + \frac{\partial T}{\partial t}^{\nu} \left(\rho_o \frac{\partial h_o}{\partial T} - \rho_v \frac{\partial h_v}{\partial T} \right) \right]$$
$$- \phi \frac{\partial P_{c_{ow}}}{\partial S_o} \frac{\partial S_o}{\partial t}^{\nu} - P_{c_{ow}} (1-\phi) \left(\frac{\partial (\vec{\nabla} \cdot \mathbf{u})}{\partial t}^{\nu} + \frac{1}{\rho_r} \frac{\partial \rho_r}{\partial T} \frac{\partial T}{\partial t}^{\nu} \right) \quad \text{(A.39)}$$

$$D_6 = -(1-\phi) \left[\frac{\partial (\vec{\nabla} \cdot \mathbf{u})}{\partial t}^{\nu} + \frac{1}{\rho_r} \frac{\partial \rho_r}{\partial T} \frac{\partial T}{\partial t}^{\nu} \right] \quad \text{(A.40)}$$

$$\mathbf{D}_7 = -\mathbf{k} \left[\rho_o \lambda_o \frac{\partial P_{c_{ow}}}{\partial S_o} \left(\mathbf{g} - \frac{\partial h_o}{\partial p_w} \nabla p_w^{\nu} - \frac{\partial h_o}{\partial T} \nabla T^{\nu} \right) \right] \quad \text{(A.41)}$$

$$\mathbf{D}_8 = -\mathbf{k} \cdot \left\{ \sum_i \rho_i \lambda_i \left[2 \frac{\partial h_i}{\partial p_w} \nabla p_w^{\nu} + \left(\frac{\partial h_i}{\partial p_w} \rho_i - 1 \right) \mathbf{g} + \frac{\partial h_i}{\partial T} \nabla T^{\nu} \right] + \frac{\partial h_o}{\partial p_w} \rho_o \lambda_o \frac{\partial P_{c_{ow}}}{\partial S_o} \nabla S_o^{\nu} \right\} \quad \text{(A.42)}$$

$$\mathbf{D}_9 = -\mathbf{k} \cdot \left\{ \sum_i \rho_i \lambda_i \left(\frac{\partial h_i}{\partial T} \nabla p_w^{\nu} + \frac{\partial h_i}{\partial T} \rho_i \mathbf{g} \right) + \frac{\partial h_o}{\partial T} \rho_o \lambda_o \frac{\partial P_{c_{ow}}}{\partial S_o} \nabla S_o^{\nu} \right\} \quad \text{(A.43)}$$

$$D_{10} = -\kappa \tag{A.44}$$

$$D_{11} = -(1-\phi)(p_w^\nu + S_o^\nu P_{c_{ow}}) \tag{A.45}$$

$$D_{12} = -(1-\phi)\rho_r \mathbf{g} \tag{A.46}$$

$$D_{13} = D_{13_{NC}} + D_{13_C} \tag{A.47}$$

$$\begin{aligned} D_{13_{NC}} = \phi \frac{\partial P_{c_{ow}}}{\partial S_o} \frac{\partial S_o}{\partial t}^\nu S_o^\nu - (1-\phi)(p_w^\nu + S_o^\nu P_{c_{ow}}) \left[\frac{\partial (\vec{\nabla} \cdot \mathbf{u})}{\partial t}^\nu + \frac{1}{\rho_r} \frac{\partial \rho_r}{\partial T} \frac{\partial T}{\partial t}^\nu \right] \\ -\phi \frac{\partial p_w}{\partial t}^\nu \left(S_w^\nu \rho_w \frac{\partial h_w}{\partial p_w} + S_o^\nu \rho_o \frac{\partial h_o}{\partial p_w} + (1 - S_w^\nu - S_o^\nu) \rho_v \frac{\partial h_v}{\partial p_w} \right) \\ -\phi \frac{\partial T}{\partial t}^\nu \left(S_w^\nu \rho_w \frac{\partial h_w}{\partial T} + S_o^\nu \rho_o \frac{\partial h_o}{\partial T} + (1 - S_w^\nu - S_o^\nu) \rho_v \frac{\partial h_v}{\partial T} \right) \end{aligned} \tag{A.48}$$

$$\begin{aligned} D_{13_C} = \mathbf{k} \cdot \left[\sum_i \rho_i \lambda_i \left(\frac{\partial h_i}{\partial p_w} \nabla p_w^\nu \cdot \vec{\nabla} p_w{}^\nu + \frac{\partial h_i}{\partial T} \nabla p_w^\nu \cdot \vec{\nabla} T^\nu \right) + \rho_o \lambda_o \frac{\partial P_{c_{ow}}}{\partial S_o} \left(\frac{\partial h_o}{\partial p_w} \nabla p_w{}^\nu + \frac{\partial h_o}{\partial T} \nabla T^\nu \right) \cdot \vec{\nabla} S_o{}^\nu \right] \\ + \mathbf{g} \cdot \mathbf{k} \cdot \left(\sum_i \rho_i^2 \lambda_i \right) \mathbf{g} \end{aligned} \tag{A.49}$$

and NC=non-convective and C=convective.

Appendix B

STATISTICAL ANALYSES OF ROCK AND FLUID PROPERTIES

The development of correlations for physical properties of water, steam, bitumen, oil, and rock is presented in this Appendix. The biggest group of properties are those of saturated steam, which are presented first.

B.1 *Correlations for saturated steam properties*

Current methods for the determination of saturated steam properties include looking up and interpolating in steam tables, and the use of discontinuous polynomial approximations to the experimental data, with heavy computational overhead over restricted regions of the saturation envelope. This appendix presents a complete suite of properties as functions of simple continuous polynomials throughout almost the entire saturation envelope.

Steam properties as a function of saturation pressure and temperature are essential for thermal reservoir simulation and other analyses. The properties are found in a computer program either by interpolation of steam table data or by using existing interpolation polynomials. The existing polynomials are of restricted range and generally discontinuous. This work offers a complete suite of steam properties as functions of continuous, simple polynomials throughout almost the entire saturation envelope—to within 7 K [13°F] or 1.85 MPa [270 psi] of the critical point of steam. Derivatives of the polynomials can be evaluated analytically.

Polynomial interpolation is necessary to make effective use of vector and parallel computers because the table look-up and interpolation is inefficient for many points, even if a single-point evaluation is quicker this way (Dubois, 1982). It must be emphasized that all the correlations are invalid outside of the steam saturation envelope.

B.1.1 *Development of correlations*

The objective of the work was that the polynomials should describe the behaviour of the steam properties to a sufficient degree of accuracy with changing pressure and temperature within the entire saturation envelope. The polynomials should be continuous and they should involve simple integer powers of the independent variable. Integer powers are efficient arithmetic operations, unlike the evaluation of transcendental functions, which implicitly includes fractional exponents. Transcendental functions take an order of magnitude more effort to evaluate than basic arithmetic operations on a computer. This efficiency is useful particularly in microcomputers. Simple polynomials can be most effectively evaluated using the Horner expansion (Knuth, 1981).

The data were tabulated in Perry and Green (1984) for saturated water substance. The property values were corroborated by the tables published by Cooper and Le Fevre (1975). The correlations were developed by polynomial regression, using the correlation coefficient and the residuals to judge the suitability of each correlation. Two transcendental evaluations are required for the saturation temperature and for the vapour density, and the evaluation of a square root for the specific enthalpy of vaporization. The polynomials all have integer powers.

The correlations are presented jointly in derived S.I. and U.S. units, where version (a) of the equation is in S.I. units, and version (b) in U.S. units. All temperatures used as independent variables in the correlations are in K [°R], and pressures similarly used are in kPa [psia]. The coefficients were developed separately for each unit system, so that the residuals are identical for each pair of approximations. Correlations are presented first for saturated steam condensate, then for saturated steam. Note that the use of temperature as the correlation variable results in simpler expressions than in many previously published works. If pressure is required as the independent variable in these correlations, then the evaluation of temperature from Equation (B.10) is very accurate and suitable for determining all the other properties.

B. 1. 2 *Discussion of error*

The maximum and mean absolute residuals from using the correlations on the original data, expressed as percentages of the true values, are reported for the properties. The correlation coefficients are reported for the regression variables, which are sometimes transformations of the properties. The suitability of the new polynomials under particular conditions can be assessed using the residual plots in Figures (B–1) to (B–11), which correspond to each equation of the same number. The correlations have been compared to the most recently published work, and the comparison is fully documented in Tortike and Farouq Ali (1989).

B. 1. 3 *Steam condensate*

Viscosity, Pa.s [cp], valid for 273.15 K $\leq T \leq$ 645 K [492°R $\leq T \leq$ 1161°R], maximum absolute residual = 2.87%, mean absolute residual = 1.07%, and $r^2 = 0.99994$.

$$\mu_{sc} = -0.0123274 + \frac{27.1038}{T} - \frac{23527.5}{T^2} + \frac{1.01425 \times 10^7}{T^3} - \frac{2.17342 \times 10^9}{T^4} + \frac{1.86935 \times 10^{11}}{T^5} \quad \text{(B.1a)}$$

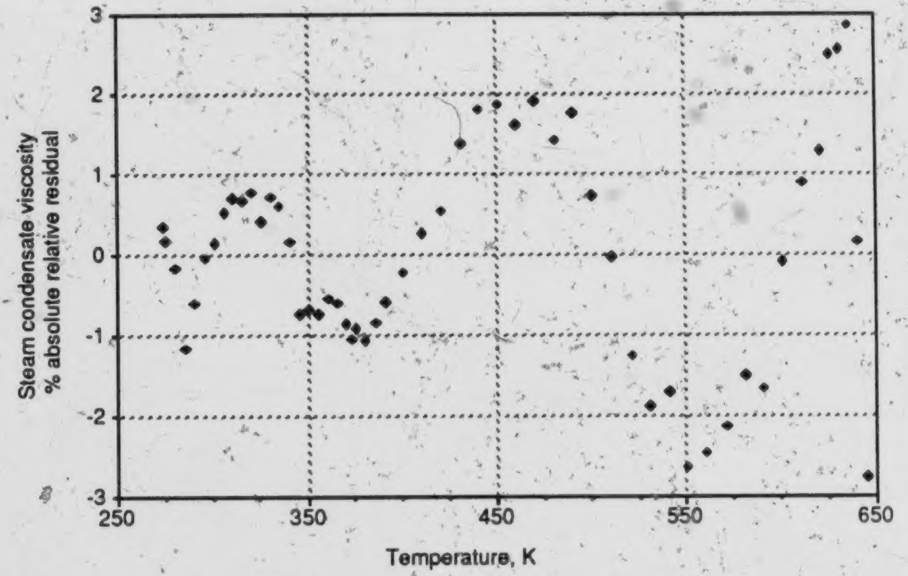


Figure B–1

Plot of residuals from the correlation of μ_{sc} with T.

$$\mu_{sc} = -12.3274 + \frac{48786.8}{T} - \frac{7.62292 \times 10^{7}}{T^{2}} + \frac{5.91509 \times 10^{10}}{T^{3}} - \frac{2.28157 \times 10^{13}}{T^{4}} + \frac{3.53226 \times 10^{15}}{T^{5}}. \qquad \text{(B.1b)}$$

Thermal conductivity, W m^{-1} K^{-1} [Btu hr^{-1} ft^{-1} $^{\circ}\text{R}^{-1}$], valid for $273.15\,\text{K} \leq T \leq 645\,\text{K}$, [$492^{\circ}\text{R} \leq T \leq 1161^{\circ}\text{R}$], maximum absolute residual = 4.58%, mean absolute residual = 0.46%, and $\text{r}^2 = 0.99857$.

$$k_{hsc} = 3.51153 - 0.0443602T + 2.41233 \times 10^{-4}T^{2} - 6.05099 \times 10^{-7}T^{3} + 7.22766 \times 10^{-10}T^{4} - 3.37136 \times 10^{-13}T^{5}. \qquad \text{(B.2a)}$$

$$k_{hsc} = 2.02892 - 0.0142394T + 4.30191 \times 10^{-5}T^{2} - 5.99485 \times 10^{-8}T^{3} + 3.97811 \times 10^{-11}T^{4} - 1.03089 \times 10^{-14}T^{5}. \qquad \text{(B.2b)}$$

Density, kg m^{-3} [lbm ft^{-3}], valid for $273.15\,\text{K} \leq T \leq 640\,\text{K}$, [$492^{\circ}\text{R} \leq T \leq 1152^{\circ}\text{R}$], maximum

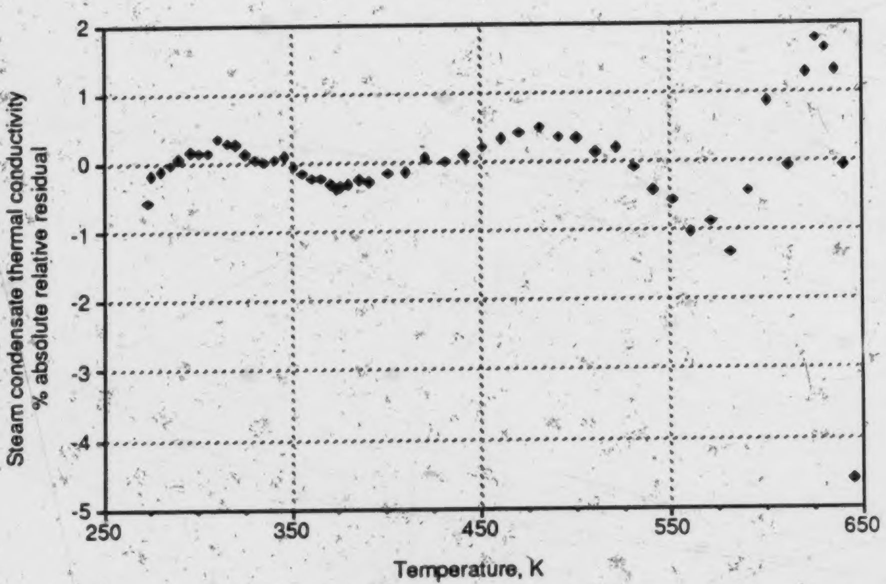


Figure B–2

Plot of residuals from the correlation of k_{hsc} with T.

absolute residual = 1.87%, mean absolute residual = 0.22%, and $r^2 = 0.99977$. N.B. all the absolute residuals are less than 1% for $273.15\,\mathrm{K} \le T < 620\,\mathrm{K}$ [$492°\mathrm{R} \le T < 1116°\mathrm{R}$].

$$\rho_{sc} = 3786.31 - 37.2487T + 0.196246T^2 - 5.04708 \times 10^{-4}T^3 + 6.29368 \times 10^{-7}T^4 - 3.08480 \times 10^{-10}T^5. \tag{B.3a}$$

$$\rho_{sc} = 236.372 - 1.29187T + 0.00378125T^2 - 5.40258 \times 10^{-6}T^3 + 3.74277 \times 10^{-9}T^4 - 1.01916 \times 10^{-12}T^5. \tag{B.3b}$$

Specific enthalpy, kJ kg^{-1} [Btu lbm^{-1}], valid for $273.15\,\mathrm{K} \le T \le 645\,\mathrm{K}$, [$492°\mathrm{R} \le T \le 1161°\mathrm{R}$], maximum absolute residual = 2.93%, mean absolute residual = 0.52%, and $r^2 = 0.99994$. N.B. all the absolute residuals are less than 1% for $305\,\mathrm{K} < T < 645\,\mathrm{K}$ [$549°\mathrm{R} < T < 1161°\mathrm{R}$].

$$h_{sc} = 23665.2 - 366.232T + 2.26952T^2 - 0.00730365T^3 + 1.30241 \times 10^{-5}T^4 - 1.22103 \times 10^{-8}T^5 + 4.70878 \times 10^{-12}T^6. \tag{B.4a}$$

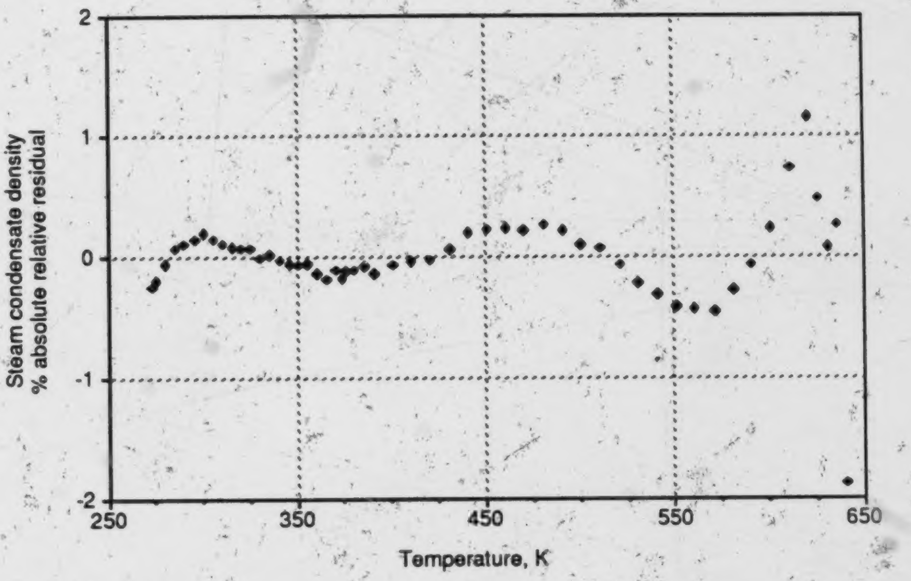


Figure B–3

Plot of residuals from the correlation of ρ_{sc} with T.

$$h_{sc} = 10174.2 - 87.4729T + 0.301147T^2 - 5.38409 \times 10^{-4}T^3 + 5.33392 \times 10^{-7}T^4 - 2.77814 \times 10^{-10}T^5 + 5.95201 \times 10^{-14}T^6. \quad \text{(B.4b)}$$

B. 1. 4 *Steam vapour*

Viscosity, Pa.s [cp], valid for 273.15 K $\leq T \leq$ 645 K, [492°R $\leq T \leq$ 1161°R], maximum absolute residual = 6.41%, mean absolute residual = 1.59%, and $r^2 = 0.99542$. Note that the errors are worst at the extremities of the range.

$$\mu_{sv} = -5.46807 \times 10^{-4} + 6.89490 \times 10^{-6}T - 3.39999 \times 10^{-8}T^2 + 8.29842 \times 10^{-11}T^3 - 9.97060 \times 10^{-14}T^4 + 4.71914 \times 10^{-17}T^5. \quad \text{(B.5a)}$$

$$\mu_{sv} = -0.546807 + 3.83050 \times 10^{-3}T - 1.04938 \times 10^{-5}T^2 + 1.42291 \times 10^{-8}T^3 - 9.49798 \times 10^{-12}T^4 + 2.49747 \times 10^{-15}T^5. \quad \text{(B.5b)}$$

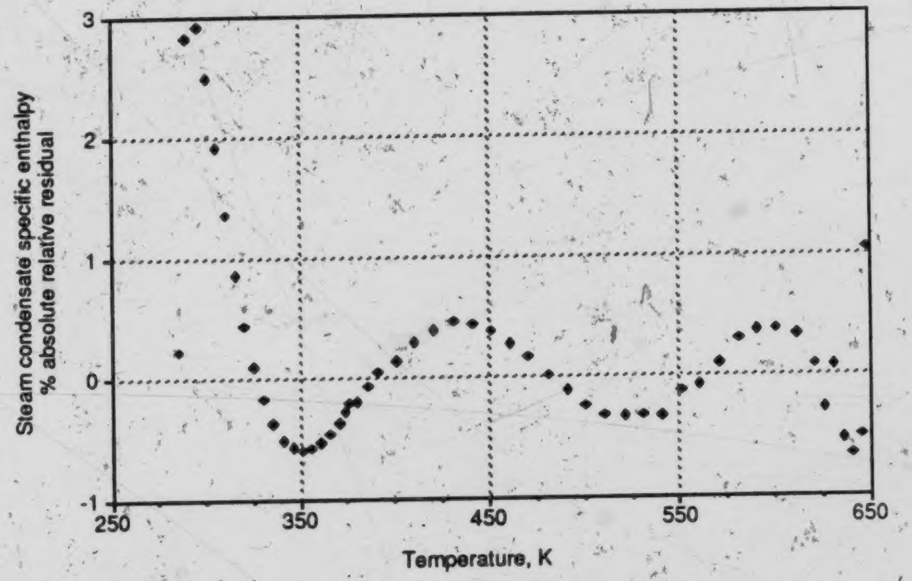


Figure B–4

Plot of residuals from the correlation of h_{sc} with T.

Thermal conductivity, W $\mathrm{m^{-1}\,K^{-1}}$ [Btu $\mathrm{hr^{-1}\,ft^{-1}\,{}^\circ R^{-1}}$], valid for $273.15\,\mathrm{K} \le T \le 645\,\mathrm{K}$, [$492^\circ\mathrm{R} \le T \le 1161^\circ\mathrm{R}$], maximum absolute residual = 7.56%, mean absolute residual = 2.63%, and $r^2 = 0.99709$. Note that the errors are worst at the extremities of the range.

$$k_{hsv} = -2.35787 + 0.0297429T - 1.46888 \times 10^{-4}T^2 + 3.57767 \times 10^{-7}T^3 - 4.29764 \times 10^{-10}T^4 + 2.04511 \times 10^{-13}T^5. \quad \text{(B.6a)}$$

$$k_{hsv} = -1.36235 + 9.54729 \times 10^{-3}T - 2.61945 \times 10^{-5}T^2 + 3.54448 \times 10^{-8}T^3 - 2.36542 \times 10^{-11}T^4 + 6.25351 \times 10^{-15}T^5. \quad \text{(B.6b)}$$

Density, kg $\mathrm{m^{-3}}$ [lbm $\mathrm{ft^{-3}}$], valid for $273.15\,\mathrm{K} \le T \le 645\,\mathrm{K}$, [$492^\circ\mathrm{R} \le T \le 1161^\circ\mathrm{R}$], maximum absolute residual = 7.71%, mean absolute residual = 1.29%, and $r^2 = 0.99996$ on $\ln \rho_{sv}$. Note that one can find the steam density using the real gas formula $\rho = pM/(zRT)$ assuming that the z-factor

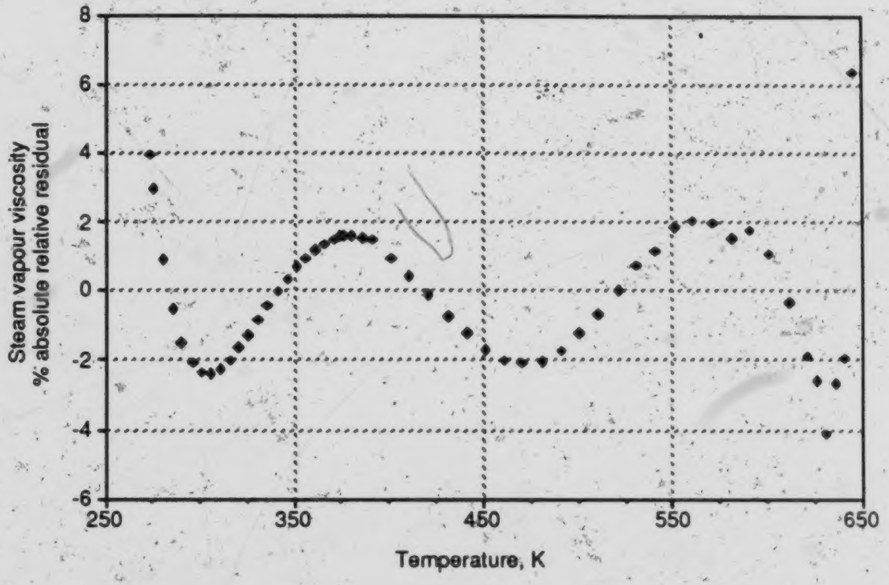


Figure B–5

Plot of residuals from the correlation of μ_{sv} with T.

has been evaluated already.

$$\ln \rho_{sv} = -93.7072 + 0.833941T - 0.003208097T^2 + 6.57652 \times 10^{-6}T^3 - 6.93747 \times 10^{-9}T^4 + 2.97203 \times 10^{-12}T^5. \quad \text{(B.7a)}$$

$$\ln \rho_{sv} = -96.4809 + 0.463301T - 9.90153 \times 10^{-4}T^2 + 1.12766 \times 10^{-6}T^3 - 6.60862 \times 10^{-10}T^4 + 1.57286 \times 10^{-13}T^5. \quad \text{(B.7b)}$$

Specific enthalpy, kJ kg^{-1} [Btu lbm^{-1}], valid for 273.15 K $\leq T \leq$ 640 K, [492°R $\leq T \leq$ 1152°R], maximum absolute residual = 0.50%, mean absolute residual = 0.08%, and $r^2 = 0.99920$.

$$h_{sv} = -22026.9 + 365.317T - 2.25837T^2 + 0.007374207T^3 - 1.33437 \times 10^{-5}T^4 + 1.26913 \times 10^{-8}T^5 - 4.96880 \times 10^{-12}T^6. \quad \text{(B.8a)}$$

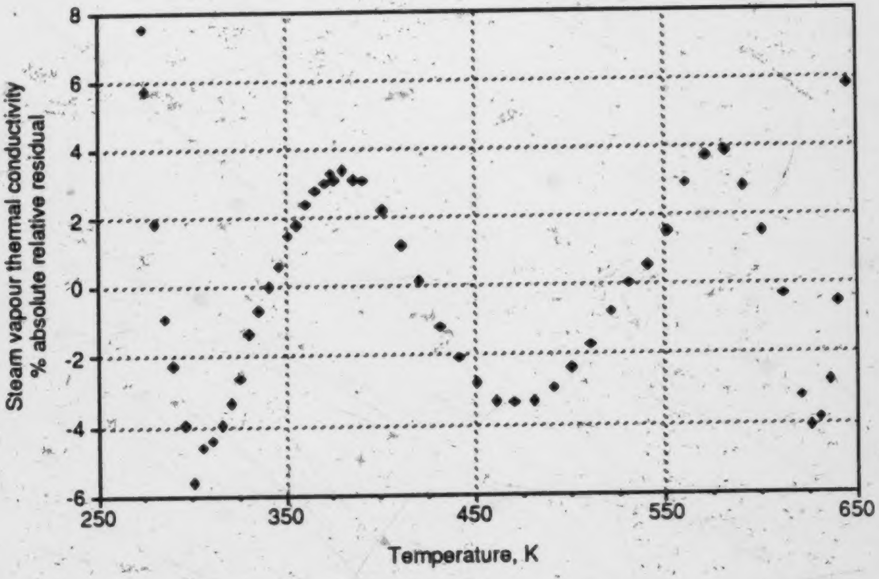


Figure B–6

Plot of residuals from the correlation of k_{hsv} with T.

$$h_{sv} = -9469.85 + 87.2545T - 0.299668T^2 + 5.43610 \times 10^{-4}T^3 - 5.46484 \times 10^{-7}T^4$$
$$+ 2.88759 \times 10^{-10}T^5 - 6.28068 \times 10^{-14}T^6. \quad \text{(B.8b)}$$

B.1.5 *Steam vapour and condensate*

Specific enthalpy of vaporization (latent heat), kJ kg^{-1} [Btu lbm^{-1}], valid for 273.15 K $\leq T \leq$ 645 K, [492°R $\leq T \leq$ 1161°R], maximum absolute residual = 5.68%, mean absolute residual = 0.40%, and r^2 = 0.99994 on L_v^2. Note that all the absolute residuals are less than 1% for 273.15 K $\leq T <$ 620 K [492°R $\leq T \leq$ 1116°R].

$$L_v = \left(7184500 + 11048.6T - 88.4050T^2 + 0.162561T^3 - 1.21377 \times 10^{-4}T^4\right)^{\frac{1}{2}}. \quad \text{(B.9a)}$$

$$L_v = \left(1327940 + 1134.53T - 5.04327T^2 + 5.15204 \times 10^{-3}T^3 - 2.13711 \times 10^{-6}T^4\right)^{\frac{1}{2}}. \quad \text{(B.9b)}$$

Saturation temperature, K [°R], valid for 0.611 kPa $\leq p \leq$ 22.12 MPa, [0.089 psia $\leq p \leq$

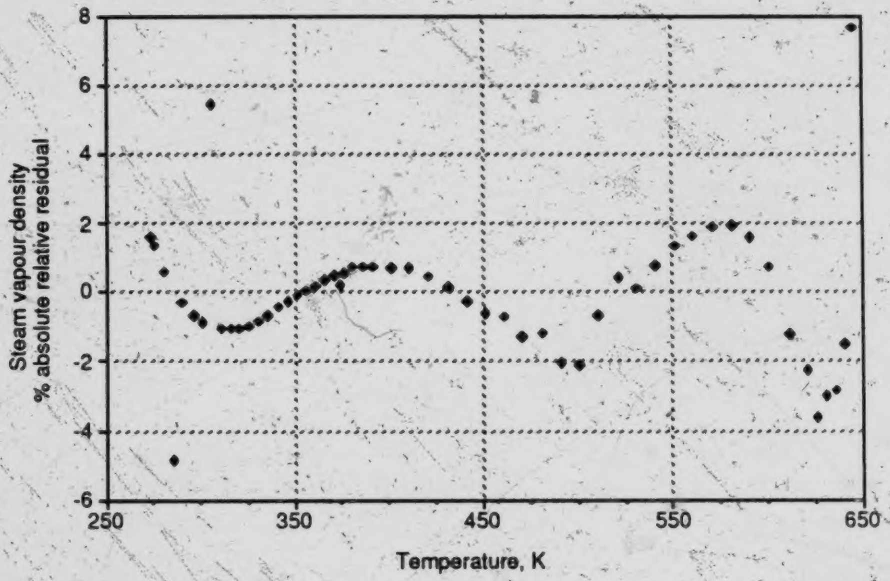


Figure B–7

Plot of residuals from the correlation of ρ_{sv} with T.

3208.2 psia], maximum absolute residual = 0.11%, mean absolute residual = 0.03%, and r^2 = 1.00000. Note that the pressure range corresponds to $273.15\,\mathrm{K} \le T \le 647.3\,\mathrm{K}$ [$491.7^\circ\mathrm{R} \le T \le 1165.1^\circ\mathrm{R}$].

$$T = 280.034 - 14.0856 \ln p + 1.38075(\ln p)^2 - 0.101806(\ln p)^3 + 0.0190017(\ln p)^4. \qquad \text{(B.10a)}$$

$$T = 561.435 + 33.8866 \ln p + 2.18893(\ln p)^2 + 0.0808998(\ln p)^3 + 0.0342030(\ln p)^4. \qquad \text{(B.10b)}$$

Saturation pressure, kPa [psia], valid for $280\,\mathrm{K} \le T \le 647.3\,\mathrm{K}$, [$504^\circ\mathrm{R} \le T \le 1165.1^\circ\mathrm{R}$], maximum absolute residual = 3.22%, mean absolute residual = 0.49%, and $r^2 = 1.00000$ on $\sqrt{p}$. Note that all the absolute residuals are less than 1% for $320\,\mathrm{K} \le T \le 647.3\,\mathrm{K}$ [$576^\circ\mathrm{R} \le T \le 1165.1^\circ\mathrm{R}$].

$$p = \Big(-175.776 + 2.29272T - 0.0113953T^2 + 2.62780 \times 10^{-5}T^3 - 2.73726 \times 10^{-8}T^4 + 1.13816 \times 10^{-11}T^5\Big)^2. \qquad \text{(B.11a)}$$

$$p = \Big(-66.9421 + 0.485086T - 1.33944 \times 10^{-3}T^2 + 1.71599 \times 10^{-6}T^3 - 9.93039 \times 10^{-10}T^4$$

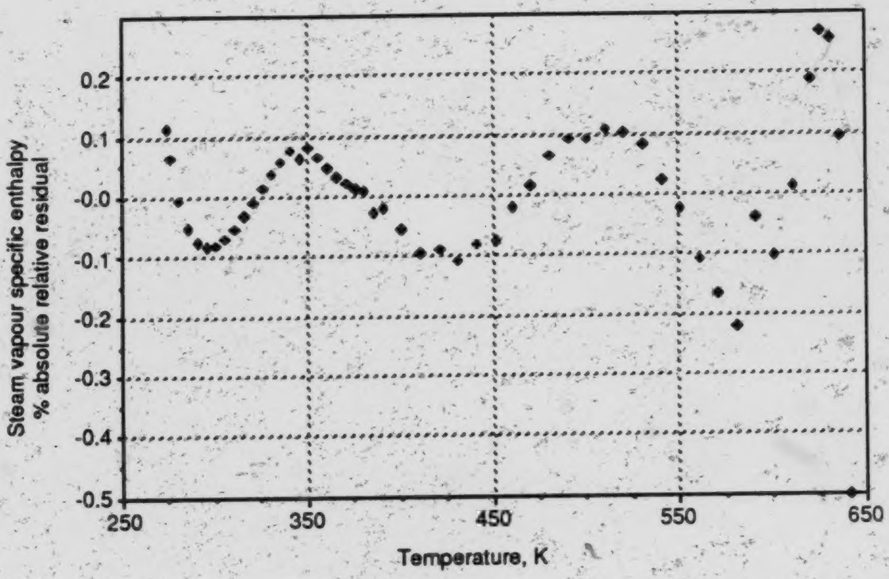


Figure B–8

Plot of residuals from the correlation of h_{sv} with T.

$$+ 2.29394 \times 10^{-13} T^{5} \Big)^{2}. \quad \text{(B.11b)}$$

B.1.6 *Conclusions — properties of saturated steam*

A new set of functional correlations has been developed to predict the physical properties of saturated steam. The correlations are valid virtually throughout the steam saturation envelope except in the immediate vicinity of the critical point. The new correlations have the following advantages over previously published work:

1. a continuous and numerically-efficient polynomial for each property throughout almost the entire steam saturation envelope, for which continuous numerical or analytical derivatives can be found;
2. suitable for vector pipeline and parallel processors, and convenient for computer spreadsheets;
3. a complete selection of steam properties with the choice of derived S.I. or U.S. units, each

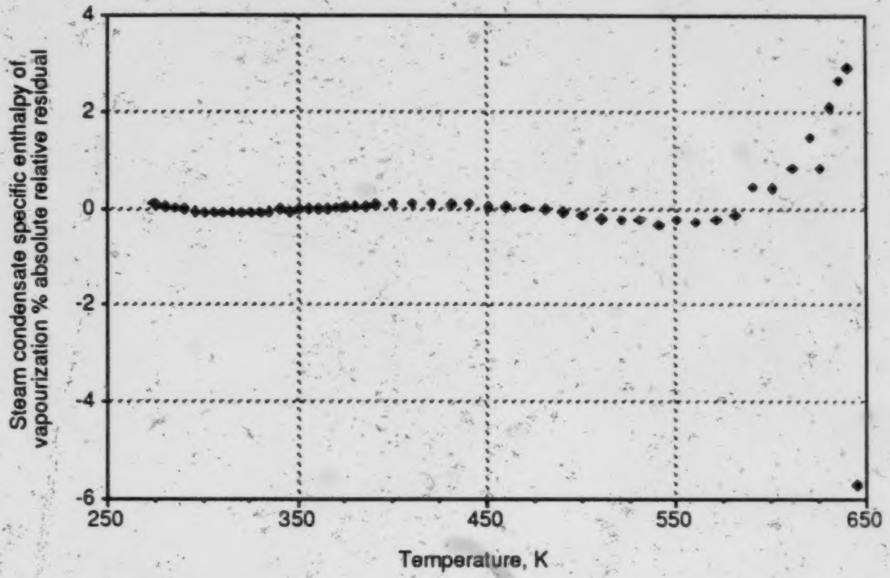


Figure B–9

Plot of residuals from the correlation of L_v with T.

correlation found separately in its own unit system.

The purpose of the correlations is to eliminate the need for different correlations over several smaller ranges, and to minimize the use of transcendental functions (which implicitly include non-integer exponents). The correlations have been compared with the previously most comprehensive set of steam property correlations in the petroleum engineering literature, and were found to have errors of similar magnitude in the restricted region of validity of the previous set.

It is of interest to note one of the first polynomial approximations for steam condensate viscosity, that by Gottfried (1965). The errors in this approximation are shown in Figure B–12. While the errors are much worse at the extremities of the range than in the correlation given here, Gottfried's correlation works well in the middle of the range and is useful because of its simple form:

$$\mu_{sc} = \frac{1242.4 - T}{26.5T - 6822} \times 10^{-3} \tag{B.12}$$

The viscosity is given here in Pa.s and the temperature in K.

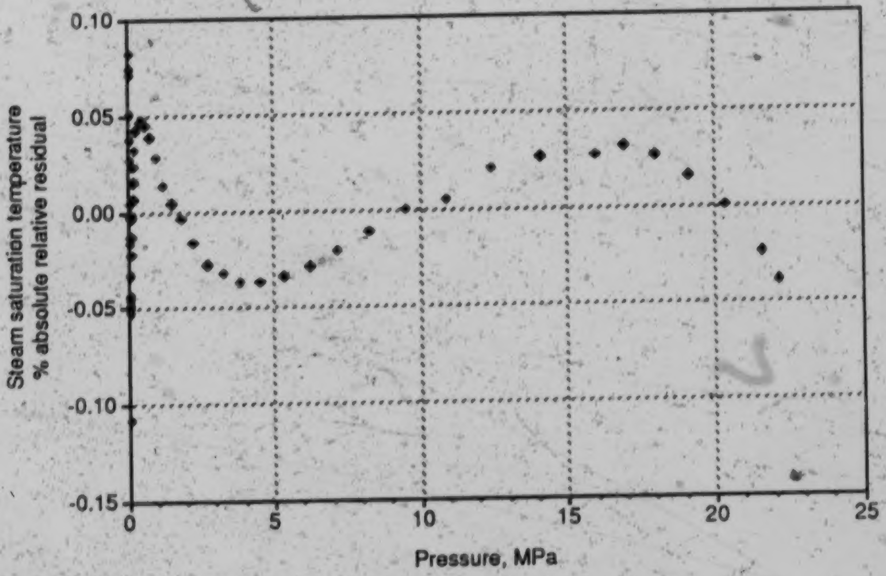


Figure B–10

Plot of residuals from the correlation of T with p.

B. 1. 7 *Bitumen*

Enthalpy, J kg^{-1}, obtained from data generated by the Simulation Science "PROCESS" program.

$$h_b = a + bT + cT^2 + dp + ep^2. \tag{B.13}$$

$$273\,\mathrm{K} \leq T \leq 625\,\mathrm{K} \quad \text{and} \quad 0.1\,\mathrm{MPa} \leq p \leq 21\,\mathrm{MPa}$$

The correlation information is shown in Figure B–13.

B. 1. 8 *Gas (methane)*

Solution gas ratio, vol/vol, data taken from Svrcek and Mehrotra (1982). A suitable multiple regression was chosen with some extrapolation.

$$R_{so} = a + bp + c\frac{p}{T}. \tag{B.14}$$

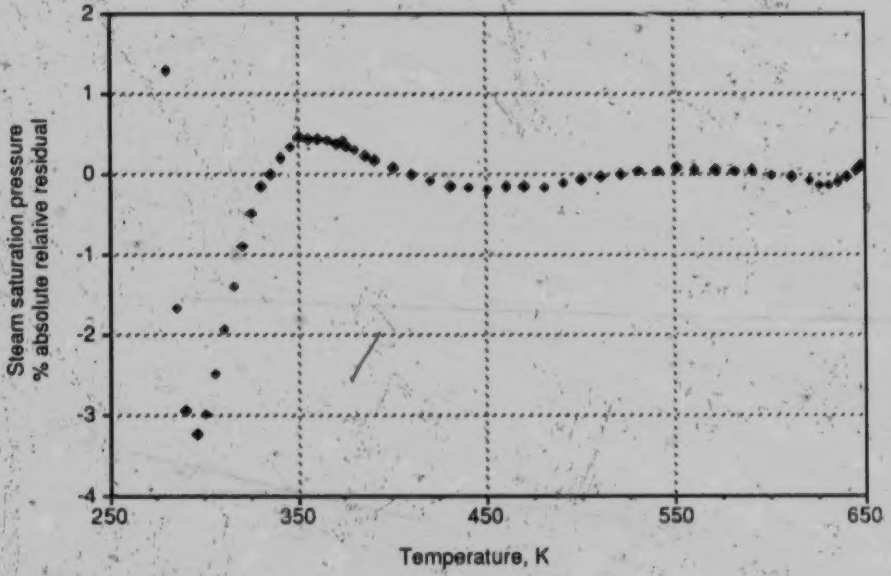


Figure B–11

Plot of residuals from the correlation of p with T.

The correlation information is shown in Figure B–14.

Viscosity, Pa.s (data taken from Perry and Green, 1984)

$$\mu_g = a + bT. \tag{B.15}$$

$$250\,\mathrm{K} \le T \le 600\,\mathrm{K}$$

The correlation information is shown in Figure B–15.

Thermal conductivity, W m^{-1} K^{-1} (data taken from Perry and Green, 1984)

$$\kappa_g = a + bT + cT^2. \tag{B.16}$$

$$250\,\mathrm{K} \le T \le 600\,\mathrm{K}$$

The correlation information is shown in Figure B–16.

Enthalpy, J kg^{-1} (data taken from Perry and Green, 1984). Several multiple regressions were

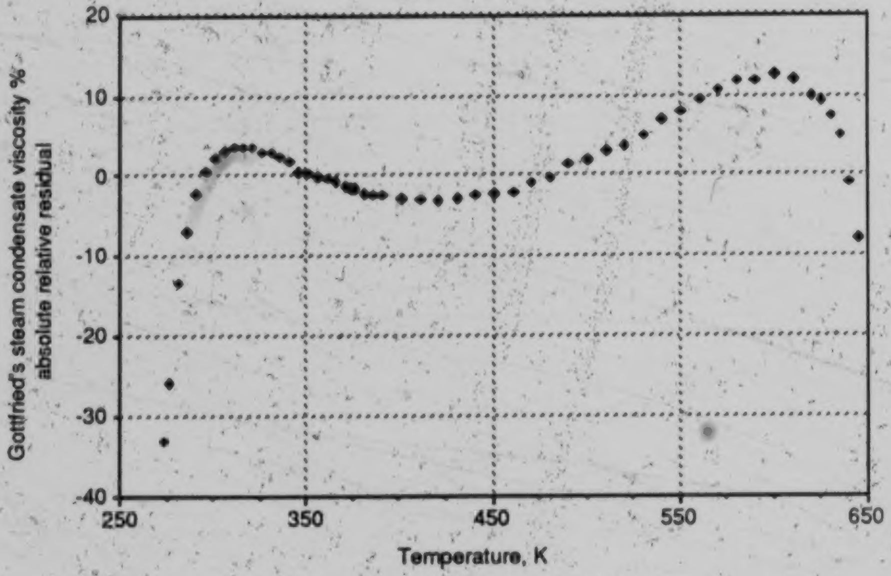
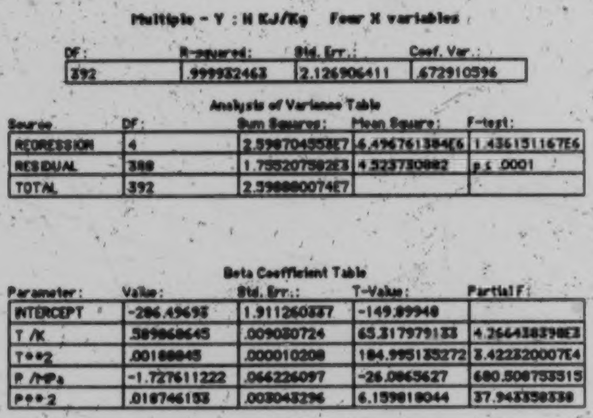


Figure B–12

Plot of residuals from Gottfried's correlation of μ_{sc} with T.



Multiple - Y : H KJ/Kg Four X variables

DF:	R-squared:	Std. Err.:	Coef. Var.:
392	.999932463	2.126906411	.672910596

Analysis of Variance Table

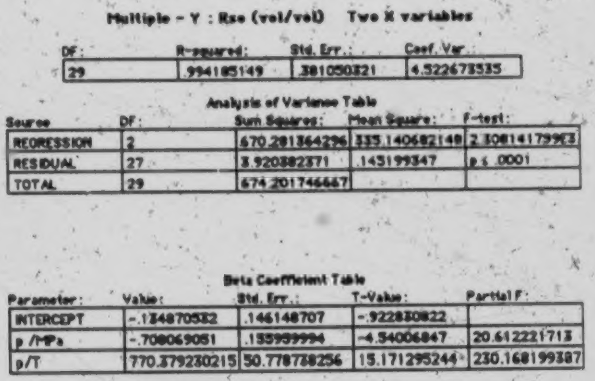
Source	DF:	Sum Squares:	Mean Square:	F-test:
REGRESSION	4	2.598704558E7	6.496761384E6	1.436151167E6
RESIDUAL	388	1.755207582E3	4.523730882	p ≤ .0001
TOTAL	392	2.598880074E7		

Beta Coefficient Table

Parameter:	Value:	Std. Err.:	T-Value:	Partial F:
INTERCEPT	-286.49695	1.911260337	-149.89948	
T /K	.589868645	.009030724	65.317979133	4.266438398E3
T**2	.00188845	.000010208	184.995135272	3.422320007E4
P /MPa	-1.727611222	.066226097	-26.0865627	680.508753315
P**2	.018746153	.003043296	6.159818044	37.943358338

Figure B–13

Enthalpy of Athabasca bitumen as a function of temperature under saturated conditions.



Multiple - Y : Rse (vol/vol)   Two X variables

DF:	R-squared:	Std. Err.:	Coef. Var.:
29	.994185149	.381050321	4.522673535

Analysis of Variance Table

Source	DF:	Sum Squares:	Mean Square:	F-test:
REGRESSION	2	670.281364296	335.140682148	2.308141799E3
RESIDUAL	27	3.920382371	.145199347	p ≤ .0001
TOTAL	29	674.201746667		

Beta Coefficient Table

Parameter:	Value:	Std. Err.:	T-Value:	Partial F:
INTERCEPT	-.134870532	.146148707	-.922830822	
p /MPa	-.708069051	.155959994	-4.54006847	20.612221713
p/T	770.379230215	50.778738256	15.171295244	230.168199387

Figure B–14

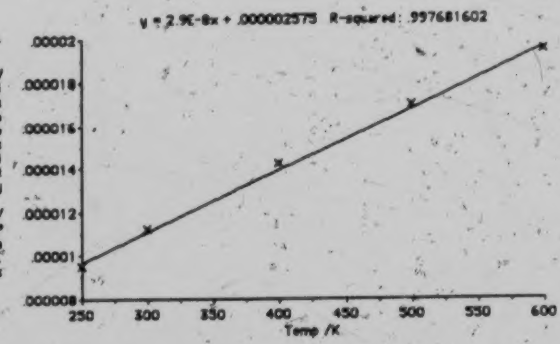
Solubility of methane in Athabasca bitumen as a function of temperature and pressure.

investigated, and the following correlation was chosen:

$$h_g = a + bT + cp + dpT. \tag{B.17}$$

$$250\,\mathrm{K} \leq T \leq 500\,\mathrm{K} \quad \text{and} \quad 0.1\,\mathrm{MPa} \leq p \leq 30\,\mathrm{MPa}$$

for which $r^2 = 0.996$. The correlation information is shown in Figure B–17.



Simple - Y : Viscosity /Pa.s X : Temp /K

DF:	R-squared:	Std. Err.:	Coef. Var.:
4	.99768	2.27526E-7	1.59332

Beta Coefficient Table

Parameter:	Value:	Std. Err.:	Variance:	T-Value:
INTERCEPT	.000002575	3.4129E-7	1.16479E-13	7.54491
SLOPE	2.85488E-8	7.94557E-10	6.31321E-19	35.93044

Analysis of Variance Table

Source	DF:	Sum Squares:	Mean Square:	F-test:
REGRESSION	1	6.68327E-11	6.68327E-11	1290.9967
RESIDUAL	3	1.55305E-13	5.17683E-14	p ≤ .0001
TOTAL	4	6.6988E-11		

Residual Information Table

SS[e(i)-e(i-1)]:	e ≥ 0:	e < 0:	DW test:
2.243E-13	3	2	1.44426

Figure B–15

Viscosity of methane as a function of temperature.

B.1.9 *Rock-fluid properties*

The relative permeability data for Athabasca tar sand was taken from Tortike (1984):

$$k_{rg} = a + bS_\ell + cS_\ell^2 + dS_\ell^3, \tag{B.18a}$$

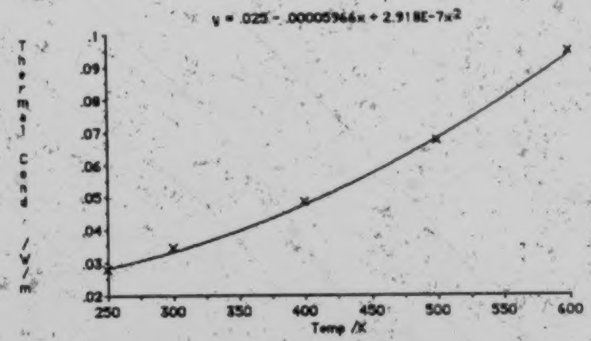
$$k_{rw} = a + bS_w + cS_w^2 + dS_w^3, \tag{B.18b}$$

$$k_{r_{ow}} = a + bS_w + cS_w^2 + dS_w^3 + eS_w^4 + fS_w^5, \tag{B.18c}$$

$$k_{r_{og}} = a + bS_\ell + cS_\ell^2 + dS_\ell^3, \tag{B.18d}$$

$$P_{c_{ow}} = a + bS_w + cS_w^2 + dS_w^{-1}, \tag{B.18e}$$

The correlation information is shown in Figures B–18, B–19, B–20, and B–21.



Polynomial - Y : Thermal Cond. /W/m-K X : Temp /K

DF:	R-squared:	Std. Err.:	Coef. Var.:
4	.999015354	.001204387	2.211506872

Analysis of Variance Table

Source	DF:	Sum Squares:	Mean Square:	F-test:
REGRESSION	2	.002943431	.001471715	1014.59329849
RESIDUAL	2	.000002901	.000001451	.005 < p ≤ .01
TOTAL	4	.002946332		

Beta Coefficient Table

Parameter:	Value:	Std. Err.:	T-Value:	Partial F:
INTERCEPT	.02508	.00729	3.44177	
x	-.00006	.00004	-1.61248	2.60008
x^2	2.91848E-7	4.3434E-8	6.71933	45.14941

Figure B–16

Thermal conductivity of methane as a function of temperature.

Multiple - Y : H kJ/Kg   Three X variables

DF:	R-squared:	Std. Err.:	Coef. Var.:
65	.996110982	15.945372315	1.199594403

Analysis of Variance Table

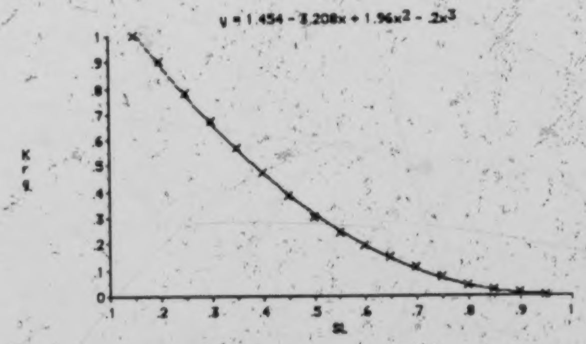
Source	DF:	Sum Squares:	Mean Square:	F-test:
REGRESSION	3	4.037650956E6	1.345883652E6	5.293442373E3
RESIDUAL	62	1.576380369E4	254.254898251	p ≤ .0001
TOTAL	65	4.053414759E6		

Beta Coefficient Table

Parameter:	Value:	Std. Err.:	T-Value:	Partial F:
INTERCEPT	431.339454574	12.327911896	34.988849549	
T /K	2.526785171	.032053907	78.829242482	6.214049473E3
P /MPa	-18.097315898	.97843372	-18.496209835	342.109778252
P*T MPa*K	.033181813	.002544034	13.042992773	170.119660485

Figure B–17

Enthalpy of methane as a function of temperature and pressure.



Polynomial - Y : Krg X : SL

DF:	R-squared:	Std. Err.:	Coef. Var.:
16	.999336342	.009411727	2.711853685

Analysis of Variance Table

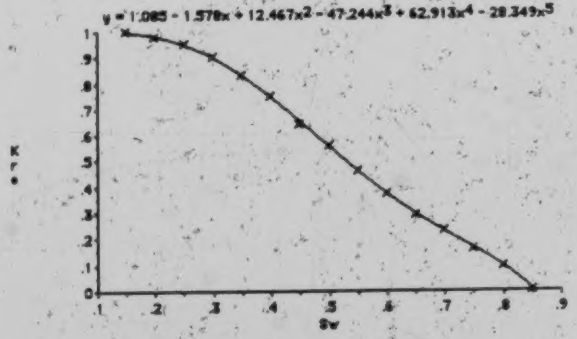
Source	DF:	Sum Squares:	Mean Square:	F-test:
REGRESSION	3	1.734001393	.578000464	6.525134964E3
RESIDUAL	13	.001151548	.000088581	p ≤ .0001
TOTAL	16	1.735152941		

Beta Coefficient Table

Parameter:	Value:	Std. Err.:	T-Value:	Partial F:
INTERCEPT	1.453916409	.024623754	59.045277514	
x	-3.208195734	.168230229	-19.070269124	363.675164475
x^2	1.960268318	.335320428	5.845955558	34.175196385
x^3	-.199518404	.201565498	-.989844023	.979791191

Figure B–18

Gas-liquid relative permeabilities.



Polynomial - Y : Kro X : Sw

DF:	R-squared:	Std. Err.:	Coef. Var.:
14	.999860008	.005098624	.931312267

Analysis of Variance Table

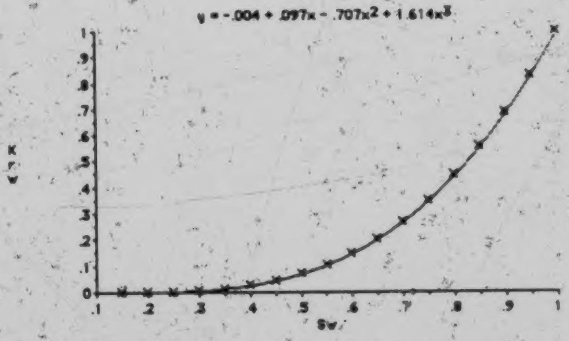
Source	DF:	Sum Squares:	Mean Square:	F-test:
REGRESSION	5	1.671033877	.334206754	1.28560991E4
RESIDUAL	9	.000233964	.000025996	p ≤ .0001
TOTAL	14	1.671267783		

Beta Coefficient Table

Parameter:	Value:	Std. Err.:	T-Value:	Partial F:
INTERCEPT	1.085092431	.084550188	12.8337081	
x	-1.577592001	1.135096897	-1.389830614	1.931629137
x^2	12.466988533	5.546114437	2.247877983	5.052955424
x^3	-47.244445647	12.510798412	-3.776293414	14.260391948
x^4	62.91252607	13.199864631	4.766329882	22.717900549
x^5	-28.348971977	5.266463452	-5.382923899	28.975869701

Figure B–19

Oil-water relative permeabilities.



Polynomial - Y : Krw X : Sw

DF:	R-squared:	Std. Err.:	Coef. Var.:
17	.999969603	.001919505	.725360381

Analysis of Variance Table

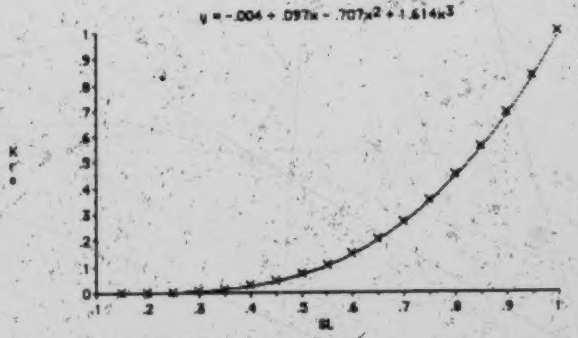
Source	DF:	Sum Squares:	Mean Square:	F-test:
REGRESSION	3	1.696925773	.565641924	1.535193311E5
RESIDUAL	14	.000051583	.000003684	p ≤ .0001
TOTAL	17	1.696977356		

Beta Coefficient Table

Parameter:	Value:	Std. Err.:	T-Value:	Partial F:
INTERCEPT	-.00369644	.004646763	-.795486945	
x	.096706828	.030556002	3.164904555	10.016620841
x^2	-.707409701	.058390598	-12.115131321	146.776406934
x^3	1.614069487	.033565296	48.08744957	2.312402806E3

Figure B–20

Water-oil relative permeabilities.



Polynomial - Y : Kro X : SL

DF:	R-squared:	Std. Err.:	Coef. Var.:
17	.999969603	.001919505	.725360381

Analysis of Variance Table

Source	DF:	Sum Squares:	Mean Square:	F-test:
REGRESSION	3	1.696925773	.565641924	1.535193311E5
RESIDUAL	14	.000051583	.000003684	p ≤ .0001
TOTAL	17	1.696977356		

Beta Coefficient Table

Parameter:	Value:	Std. Err.:	T-Value:	Partial F:
INTERCEPT	-.00369644	.004646763	-.795486945	
x	.096706828	.030556002	3.164904555	10.016620841
x^2	-.707409701	.058390598	-12.115131321	146.776406934
x^3	1.614069487	.033565296	48.08744957	2.312402806E3

Figure B–21

Oil-gas relative permeabilities.

Appendix C

THE METHOD OF WEIGHTED RESIDUALS AND FINITE ELEMENTS

Some extra material is included because of the general unfamiliarity of the petroleum engineering profession with the subject. Sections C.1 to C.5 are condensed from Zienkiewicz and Morgan (1983), Chapters 2, 3, 4 and 5, but can be found elsewhere. The remainder of the appendix is more directly related to the topic of the thesis.

C.1 *Approximation of functions*

The topic is introduced by describing the approximation of a function ϕ with a trial function, also known as a basis or shape function. Once an approximation for ϕ has been found within a given region Ω, an approximate value for ϕ can be evaluated for every point within Ω. This is in contrast to the finite difference method, where the values of ϕ are sought at a finite number of points within Ω.

Suppose that the function ϕ must also satisfy certain boundary conditions on the curve Γ enclosing the region Ω, then the approximations ought to satisfy these boundary conditions exactly. This requirement is frequently needed for the solution of differential equations. Find a function ψ which has the same values as ϕ on Γ, i.e., $\psi|_\Gamma = \phi|_\Gamma$, and find a set of independent trial functions $N_m\{m = 1, 2, 3, \ldots, M\}$ such that $N_m|_\Gamma = 0$, then ϕ can be approximated by $\hat{\phi}$ in Ω while satisfying the boundary conditions:

$$\phi \approx \hat{\phi} = \psi + \sum_{m=1}^{M} a_m N_m. \tag{C.1}$$

where $a_m\{m = 1, 2, 3, \ldots, M\}$ are parameters suitably chosen to optimize the approximation. Note that, by definition, $\hat{\phi}\Big|_\Gamma = \phi|_\Gamma$. If the set of trial functions is increased so that $M \to \infty$, then the approximation should improve so as to adequately represent ϕ. This property is known as the completeness requirement. The question now arises of how to choose the appropriate values of a_m.

C.2 *Method of weighted residuals*

The error, or residual, from the approximation to Equation (C.1) is defined by

$$R_\Omega = \phi - \hat{\phi}. \tag{C.2}$$

so that R_Ω is a function within the domain Ω. The object is to minimize R_Ω in a general manner over Ω. This is done by requiring that a number of integrals of the residual, each weighted differently, be zero. Thus we have the weighted residual statement

$$\int_\Omega W_\ell(\phi - \hat{\phi})\, d\Omega = \int_\Omega W_\ell R_\Omega \, d\Omega = 0. \quad \ell = 1, 2, 3, \ldots, M \tag{C.3}$$

$W_\ell\{\ell = 1, 2, 3, \ldots, M\}$ is a set of independent weighting functions. Thus the completeness requirement will be met if Equation (C.3) is satisfied as $M \to \infty$.

Several choices may be made for the weighting function, but the Galerkin method is most common. This method requires that the trial functions, used to approximate the function ϕ, be used as the weighting function, so $W_\ell = N_\ell$.

C. 3 *Approximation to the solution of linear partial differential equations*

Consider a general partial differential equation:

$$A(\phi) = \mathfrak{L}\phi + p = 0 \quad \text{in } \Omega. \tag{C.4}$$

where $\mathfrak{L}$ is a linear differential operator and p is independent of ϕ. Equation (C.4) must be solved subject to the boundary conditions,

$$B(\phi) = \mathfrak{M}\phi + r = 0 \quad \text{on } \Gamma. \tag{C.5}$$

where $\mathfrak{M}$ is a linear differential operator and r is independent of ϕ, and can represent Dirichlet, Neuman and mixed (Cauchy) boundary conditions simultaneously. One can use the techniques outlined at the beginning to construct an approximation $\hat{\phi}$ to the true solution ϕ:

$$\phi \approx \hat{\phi} = \psi + \sum_{m=1}^{M} a_m N_m. \tag{C.1}$$

The function ψ and the trial functions N_m are chosen so that

$$\left.\begin{aligned} \mathfrak{M}\phi &= -r \\ \mathfrak{M}N_m &= 0 \quad m = 1, 2, \ldots, M \end{aligned}\right\} \text{ on } \Gamma. \tag{C.6}$$

Now the boundary conditions are are automatically satisfied for all a_m. Actual derivatives are found by differentiation of the approximation:

$$\frac{\partial \phi}{\partial x} \approx \frac{\partial \hat{\phi}}{\partial x} = \frac{\partial \psi}{\partial x} + \sum_{m=1}^{M} a_m \frac{\partial N_m}{\partial x}. \tag{C.7}$$

and similarly for higher order derivatives. Note that because the approximation automatically satisfies the boundary conditions, it is only necessary that $\hat{\phi}$ approximately satisfies Equation (C.4)

$$R_\Omega \equiv A(\hat{\phi}) = \mathfrak{L}\hat{\phi} + p = \mathfrak{L}\psi + \left(\sum_{m=1}^{M} a_m \mathfrak{L} N_m\right) + p. \tag{C.8}$$

The method of weighted residuals can be applied to make $\hat{\phi}$ a satisfactory approximation to ϕ by choosing an appropriate set of weighting functions W_ℓ and requiring

$$\int_\Omega W_\ell R_\Omega \, d\Omega \equiv \int_\Omega W_\ell \left\{ \mathfrak{L}\psi + \left(\sum_{m=1}^{M} a_m \mathfrak{L} N_m\right) + p \right\} d\Omega = 0. \tag{C.9}$$

There are M unknowns a_m, and by applying this equation for $\ell = 1, 2, \ldots, M$ we have the set of M linear equations

$$K_{\ell m} a_m = f_\ell. \tag{C.10}$$

where the repeated subscripts indicate summation and

$$K_{\ell m} = \int_\Omega W_\ell \mathfrak{L} N_m \, d\Omega. \quad 1 \leq (\ell, m) \leq M. \tag{C.11a}$$

$$f_\ell = -\int_\Omega W_\ell p \, d\Omega - \int_\Omega W_\ell \mathfrak{L}\psi \, d\Omega. \quad 1 \leq \ell \leq M. \tag{C.11b}$$

There is also the case where the approximation function $\hat{\phi}$ is constructed so that there is no functional term ψ to automatically solve the boundary conditions:

$$\phi \approx \hat{\phi} = \sum_{m=1}^{M} a_m N_m. \tag{C.12}$$

Now one must satisfy the residuals from the approximations to the solution in the domain Ω and on the boundary Γ.

$$R_\Omega = A(\hat{\phi}) = \mathfrak{L}\hat{\phi} + p \quad \text{in } \Omega. \tag{C.13a}$$

$$R_\Gamma = B(\hat{\phi}) = \mathfrak{M}\hat{\phi} + r \quad \text{on } \Gamma. \tag{C.13b}$$

and the weighted residual statement becomes

$$\int_\Omega W_\ell R_\Omega \, d\Omega + \int_\Gamma \bar{W}_\ell R_\Gamma \, d\Gamma = 0. \tag{C.14}$$

where W_ℓ and $\bar{W}_\ell$ represent two sets of independent weighting functions, each of which is not necessarily independent of the other set. Again as $M \to \infty$, the approximation $\hat{\phi}$ tends to the exact solution ϕ.

In some circumstances it is possible to avoid the approximation of the boundary derivatives which would otherwise be difficult without going to a large value of M. The first term of Equation (C.14) is given by

$$\int_{\Omega} W_{\ell} R_{\Omega}\, d\Omega \equiv \int_{\Omega} W_{\ell} \left(\mathfrak{L}\hat{\phi} + p\right) d\Omega. \tag{C.15}$$

of which the first part can often be rearranged (e.g., by integration by parts) to obtain the form

$$\int_{\Omega} W_{\ell} \mathfrak{L}\hat{\phi}\, d\Omega \equiv \int_{\Omega} (\mathfrak{C} W_{\ell})(\mathfrak{D}\hat{\phi})\, d\Omega + \int_{\Gamma} W_{\ell} \mathfrak{E}\hat{\phi}\, d\Gamma. \tag{C.16}$$

where $\mathfrak{C}$, $\mathfrak{D}$ and $\mathfrak{E}$ are linear differential operators with orders of differentiation lower than that of the original differential operator $\mathfrak{L}$. This expression is called the *weak* form of this particular weighted residual statement.

By judicious choice of the boundary weighting function $\bar{W}_{\ell}$ in Equation (C.14), the last term in the same equation might cancel with the last term in Equation (C.16). A boundary condition which is amenable to this treatment is called *natural*. Natural boundary conditions are a subset of the Neuman conditions, which are boundary conditions on derivatives. The resultant lower orders of differentiation of the trial functions confers benefits which will be apparent in the next section.

C.4 *The finite element method and piecewise-defined trial functions*

Thus far it was implicit that the trial functions in the approximation were expressions valid everywhere in the domain Ω, and that the integrals were evaluated throughout Ω. It is possible, instead, to divide the domain Ω into nonoverlapping ***subdomains*** or ***elements*** Ω^e and then construct the approximation in a piecewise manner over each subdomain. Thus the approximation is only necessarily continuous within each subdomain, and a different set of trial functions N_m can be used in each of the subdomains Ω^e. The definite integrals arising from the weighted residual statement are found by summing the integrals over each subdomain:

$$\int_{\Omega} W_{\ell} R_{\Omega}\, d\Omega = \sum_{e=1}^{E} \int_{\Omega^e} W_{\ell} R_{\Omega}\, d\Omega; \tag{C.17a}$$

$$\int_{\Gamma} W_{\ell} R_{\Gamma}\, d\Gamma = \sum_{e=1}^{E} \int_{\Gamma^e} W_{\ell} R_{\Gamma}\, d\Gamma. \tag{C.17b}$$

where $\sum_{e=1}^{E} \Omega^e = \Omega$ and $\sum_{e=1}^{E} \Gamma^e = \Gamma$. The domain has been subdivided into E elements. Summations involving Γ^e are only relevant where a portion of the boundary Γ lies on a subdomain Ω^e.

The advantage of this approach is that subdomains of regular and simple shape are chosen, with trial functions found in a repeatable manner over the elements. Thus, a region of complex shape can be reduced to a collection of simpler geometries which can be handled easily. The sets of trial functions are also chosen to be zero outside of a given subdomain.

The derivatives of the piecewise-continuous trial functions will be necessary for the solution of partial differential equations, so certain continuity requirements must be met. The next question is how does a shape function behave at the junction of two elements? If it is discontinuous, then the derivative will be infinite, causing the integral to be most likely indeterminate. Therefore, to ensure continuity, trial functions should be chosen that do not have infinite derivatives in the integral terms of Equation (C.14). Thus if s is the highest order of derivative in an integral in the weighted residual statement, the trial function should have derivatives which are continuous to the order of at least $(s-1)$. This is referred to as functions which show C^{s-1} continuity. This continuity requirement is also applicable to the weighting functions W_ℓ. However, an exception is made for the Dirac delta function.

It was shown that the weak formulation of a weighted residual statement, e.g., Equation (C.16), contained lower order derivatives than the original formulation, e.g., Equation (C.14). Thus the weak formulation will require less strict continuity from the trial functions, i.e., a lower order of continuity. This is obviously an advantage. Note, though, that the weak formulation took the weighting functions under one of the differential operators, thus requiring that the weighting functions meet a certain order of continuity. As the trial functions have already been specified to meet the required continuity, the Galerkin method of setting $W_\ell = N_\ell$ is often convenient as $\mathfrak{C}$ and $\mathfrak{D}$ are usually operators involving the same order of differentiation. It should be stressed that while this is true in the majority of cases, it is not always so.

C.5 *Mapping and isoparametric elements*

The problem with spatial integration is that the evaluation of the integral becomes too complex for all but the simplest element geometries. This restriction could be removed by confining the integration to a simple shape with a local coordinate basis (ξ, η, ζ) which is then mapped to the more complex domain in the global coordinate system (x, y, z) where the solution is required. A mapping, by definition, describes a one-to-one relationship between each point in the (ξ, η, ζ) coordinate system to a uniquely corresponding point in the (x, y, z) coordinate system. An example is given in Figure C–1 of a linear mapping between a square element with a local coordinate system (η, ζ) originating

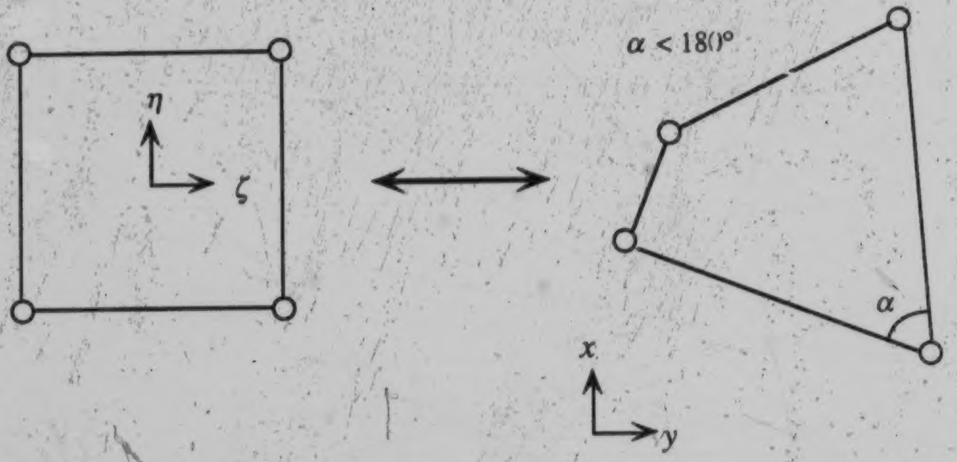


Figure C–1

Example of a linear mapping between local and global coordinate systems.

at the centre of the element, and an arbitrary quadrilateral in the global (x, y) space.

The mapping is given by a functional relationship, in general,

$$
\begin{aligned}
x &= f_1(\xi, \eta, \zeta);\\
y &= f_2(\xi, \eta, \zeta); \qquad\qquad (C.18)\\
z &= f_3(\xi, \eta, \zeta).
\end{aligned}
$$

Given a mapping, a trial function can be chosen within the local (ξ, η, ζ) coordinate system to satisfy the interelement continuity. The functional variation over the global element in (x, y, z) space is automatically covered through the mapping functions, as is the continuity in the (x, y, z) space.

Transformation rules must be applied to many terms in the integrals from the weighted residual statement, because the domain of interest is the global (x, y, z) space. Two areas must be covered: spatial derivatives and the variables of integration, both of which are part of the (x, y, z) space. Spatial derivatives of functions described in the local coordinate system are found in the (x, y, z) space using the Jacobian matrix of the transformation, J. This is found using the chain rule on the trial function, $N_\ell^e(\xi,\eta,\zeta)$, in the local element:

$$\begin{aligned}\frac{\partial N_\ell^e}{\partial \xi} &= \frac{\partial N_\ell^e}{\partial x}\frac{\partial x}{\partial \xi} + \frac{\partial N_\ell^e}{\partial y}\frac{\partial y}{\partial \xi} + \frac{\partial N_\ell^e}{\partial z}\frac{\partial z}{\partial \xi};\\ \frac{\partial N_\ell^e}{\partial \eta} &= \frac{\partial N_\ell^e}{\partial x}\frac{\partial x}{\partial \eta} + \frac{\partial N_\ell^e}{\partial y}\frac{\partial y}{\partial \eta} + \frac{\partial N_\ell^e}{\partial z}\frac{\partial z}{\partial \eta};\\ \frac{\partial N_\ell^e}{\partial \zeta} &= \frac{\partial N_\ell^e}{\partial x}\frac{\partial x}{\partial \zeta} + \frac{\partial N_\ell^e}{\partial y}\frac{\partial y}{\partial \zeta} + \frac{\partial N_\ell^e}{\partial z}\frac{\partial z}{\partial \zeta}.\end{aligned} \tag{C.19}$$

This can be written as

$$\begin{Bmatrix}\dfrac{\partial N_\ell^e}{\partial \xi}\\ \dfrac{\partial N_\ell^e}{\partial \eta}\\ \dfrac{\partial N_\ell^e}{\partial \zeta}\end{Bmatrix} = \mathbf{J}\begin{Bmatrix}\dfrac{\partial N_\ell^e}{\partial x}\\ \dfrac{\partial N_\ell^e}{\partial y}\\ \dfrac{\partial N_\ell^e}{\partial z}\end{Bmatrix}. \tag{C.20}$$

The left-hand side of this expression is easily found, as is the Jacobian matrix. Thus an expression for the spatial derivatives in the (x,y,z) space is found, provided that $\mathbf{J}$ is not singular,

$$\begin{Bmatrix}\dfrac{\partial N_\ell^e}{\partial x}\\ \dfrac{\partial N_\ell^e}{\partial y}\\ \dfrac{\partial N_\ell^e}{\partial z}\end{Bmatrix} = \mathbf{J}^{-1}\begin{Bmatrix}\dfrac{\partial N_\ell^e}{\partial \xi}\\ \dfrac{\partial N_\ell^e}{\partial \eta}\\ \dfrac{\partial N_\ell^e}{\partial \zeta}\end{Bmatrix}. \tag{C.21}$$

The variables and limits of integration must be changed to facilitate evaluation of the integral. It can be shown that

$$dx\,dy\,dz = |\mathbf{J}|\,d\xi\,d\eta\,d\zeta. \tag{C.22}$$

where $|\mathbf{J}|$ is the determinant of the Jacobian of the transformation. Typically, the local coordinate system is defined to give integration limits of -1 and $+1$, which is convenient for numerical quadrature.

A convenient form of mapping is the parametric kind, where the mapping function is similar to the trial functions. The mapping function can then be written in terms of $N_\ell^e(\xi,\eta,\zeta)$ where

$$\begin{aligned} x &= N_\ell^e x_\ell;\\ y &= N_\ell^e y_\ell;\\ z &= N_\ell^e z_\ell.\end{aligned} \tag{C.23}$$

and the summation is over repeated indices, and x_ℓ, y_ℓ, and z_ℓ represent known values at specified points. Thus the mapping will have the same continuity as the shape function. In this case, where the mapping and shape functions are identical, the mapping is called isoparametric and finite elements using this mapping are called isoparametric finite elements. Limitations do exist on the mapping.

For example, interior angles of a quadrilateral should not exceed 180°, as was stipulated in Figure C–1.

C.6 *Upwinding*

The weighting functions in the weighted residual statement must be changed when more influence must be assigned to upstream nodes. A separate, asymmetric weighting function will be used which is built from a one-dimensional analogue. Convective and non-convective operators can be separated out in the weighted residual statment, thus:

$$\int_{\Omega} \mathfrak{D}_{nc} \hat{\psi} W_i \, d\Omega + \int_{\Omega} \mathfrak{D}_c \hat{\psi} W_i^* \, d\Omega = 0. \tag{C.24}$$

where W_i^* represents the set of asymmetric weighting functions, and the subscripts c and nc mean convective and non-convective, respectively.

The one-dimensional analogue of the asymmetric weighting function will be a quadratic function, as outlined by Huyakorn (1976). Figure C–2 illustrates the x-coordinate divided into a finite element, defined by two nodes separated by a distance h,

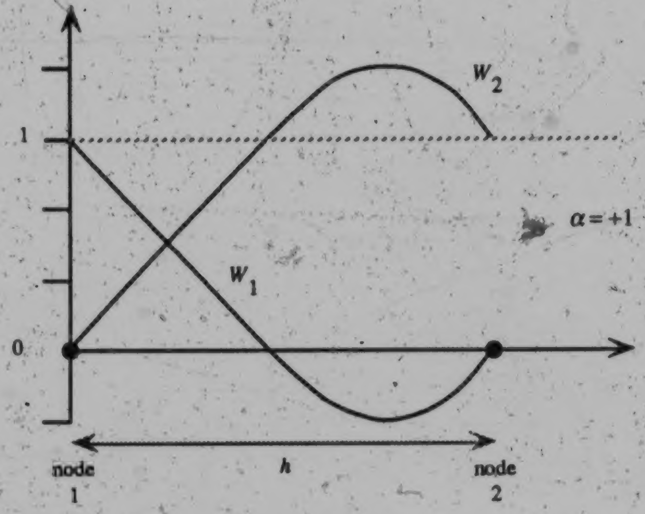


Figure C–2

Asymmetric weighting functions in a one dimensional element.

The function is found by adjusting the trial function:

$$\left.\begin{aligned} W_1^*(x) &= N_1(x) - F(x); \\ W_2^*(x) &= N_2(x) + F(x); \\ F(x) &= a\frac{x^2}{h^2} + b\frac{x}{h} + c, \end{aligned}\right\} \quad 0 \leq x \leq h. \tag{C.25}$$

where $F(x)$ is a piecewise-defined quadratic function, and a, b, and c are undetermined coefficients, h is the distance between the nodes, and x is measured from node 1. The function $F(x)$ is chosen such that $F(0) = F(h) = 0$, so that the weighting function always equals the trial function at the nodes. This constraint reduces the number of undetermined coefficients in $F(x)$ to one: $\alpha = -{}^{x}/3$, as $c = 0$ and $b = -a$. Then,

$$F(x) = 3\alpha\left(-\frac{x^2}{h^2} - \frac{x}{h}\right). \tag{C.26}$$

Thus the one-dimensional weighting functions become:

$$\left.\begin{aligned} W_1^*(x) &= N_1(x) + 3\alpha\frac{x^2}{h^2} - 3\alpha\frac{x}{h}; \\ W_2^*(x) &= N_2(x) - 3\alpha\frac{x^2}{h^2} + 3\alpha\frac{x}{h}. \end{aligned}\right\} \tag{C.27}$$

For the case of linear one-dimensional trial functions,

$$N_1 = 1 - \frac{x}{h}; \quad N_2 = 1 + \frac{x}{h}.$$

the difference between the asymmetric weighting functions and the linear trial functions can be seen in Figure C–3 for $\alpha = 1$.

When α is positive, more weighting is given to the upstream portion of the line segment. In fact, for $\alpha = 1$:

$$\left.\begin{aligned} &\int_{x_{i-1}}^{x} W_i^*(x)\,dx = h \quad x_{i-1} \leq x \leq x_i \\ &\int_{x}^{x_{i+1}} W_i^*(x)\,dx = 0 \quad x_i \leq x \leq x_{i+1} \end{aligned}\right\} \tag{C.28}$$

Thus, any property multiplied by W_i and integrated from i to $i+1$ will be weighted less than in the same integration from $i-1$ to i.

C.6.1 *Asymmetric weighting functions for linear quadrilateral elements*

The weighting function can be found at a node by taking the appropriate product of the one-dimensional weighting functions. This is also the approach taken for a three-dimensional brick

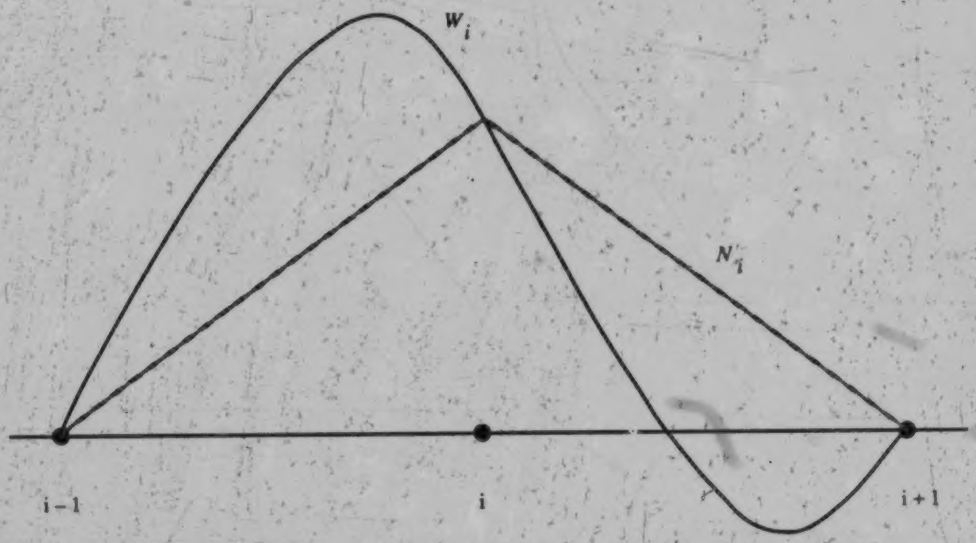


Figure C–3

Linear trial functions in one dimensional element.

element. A damping factor is required for each side, analogous to the parameter α in the one-dimensional prototype. This is illustrated in Figure C–4.

Example functions in the local ξ, η space are:

$$\left.\begin{aligned} W_1^*(\xi,\eta) &= G_1(\xi,\alpha_1)H_1(\eta,\beta_2); \\ W_3^*(\xi,\eta) &= G_2(\xi,\alpha_1)H_2(\eta,\beta_1); \end{aligned}\right\} \tag{C.29}$$

where the functions G_1, G_2, H_1, and H_2 are derived directly from the one-dimensional weighting functions in Equation (C.25). For example,

$$G_1 = 1 - \frac{x}{h} + 3\alpha_1\frac{x^2}{h^2} - 3\alpha_1\frac{x}{h}. \tag{C.30}$$

This must then be rewritten in terms of $\xi(x)$, where

$$\frac{x}{h} = \frac{\xi+1}{2}.$$

C. 6. 2 *Asymmetric weighting functions for linear triangular elements*

Again, this follows the approach taken for the one-dimensional example. The asymmetric

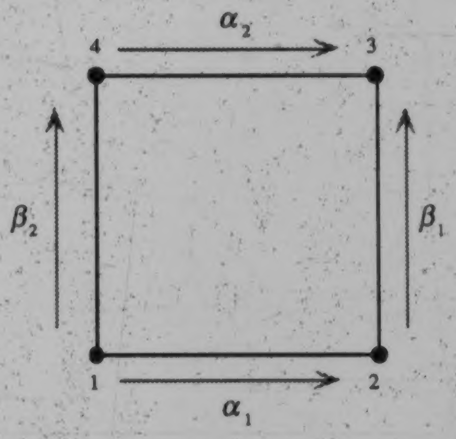


Figure C–4

Weighting parameters in quadrilateral element.

weighting functions are of the form

$$W_i^* = N_i + F_i = L_i + F_i.$$

where L_i is the set of trial functions within a triangular element. The weighting functions are shown in Figure C–5.

The F_i are quadratic functions satisfying the following:

(a) $\sum_{i=1}^{3} F_i = 1$; sum of weighting functions equals unity.
(b) $F_i(L_i = 0) = 0$; $F_i = 0$ at the side opposite node i.
(c) $|F_i(L_j = 0)| = 3\alpha_j L_k L_i$; $(i \neq j \neq k)$.

The condition (c) ensures compatibility along the sides common to the triangular and quadrilateral elements. Huyakorn (1976) suggested that the following equations be used, which do satisfy the three conditions (a), (b), and (c):

$$\left.\begin{aligned} F_1 &= 3(\alpha_2 L_3 L_1 - \alpha_3 L_2 L_1); \\ F_2 &= 3(\alpha_3 L_1 L_2 - \alpha_1 L_3 L_2); \\ F_3 &= 3(\alpha_1 L_2 L_3 - \alpha_2 L_1 L_3). \end{aligned}\right\} \qquad \text{(C.31)}$$

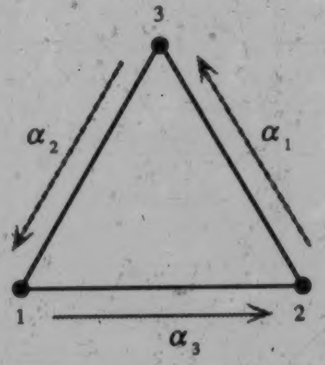


Figure C–5

Weighting parameters in triangular element.

C.7 *Shape and weighting functions of trilinear brick elements*

The trilinear element used in FESPS is the same one described by Voss (1978). It is also referred to as the L8 element because of the linear trial functions and the eight nodes characterizing the element. A triangular prism element is also used and is described in the next section. The trilinear brick element is quite common in the finite element literature, which resulted, for example, in the development of specific quadrature rules for this geometry. The L8 element is illustrated in Figure C–6.

Note the local coordinate system (ξ, η, ζ) which is centered on the centre of the element. The faces of the element are defined by a unit displacement in each direction of each of the coordinates. The traditional anti-clockwise numbering of the nodes is maintained. The weighting function parameters have directions analogous to the quadrilateral elements discussed in the previous section. The eight trilinear interpolation, or trial, functions are formed by the product of three linear one-dimensional interpolation functions. A typical linear trial function set is illustrated in Figure C–7.

In general, the linear trial function in the ξ-direction can be written

$$N_i^\xi = \frac{1}{2}(1 + \xi\xi_i). \tag{C.32}$$

where the subscript i refers to the node number, and ξ_i is the value of ξ at node i. The node numbers

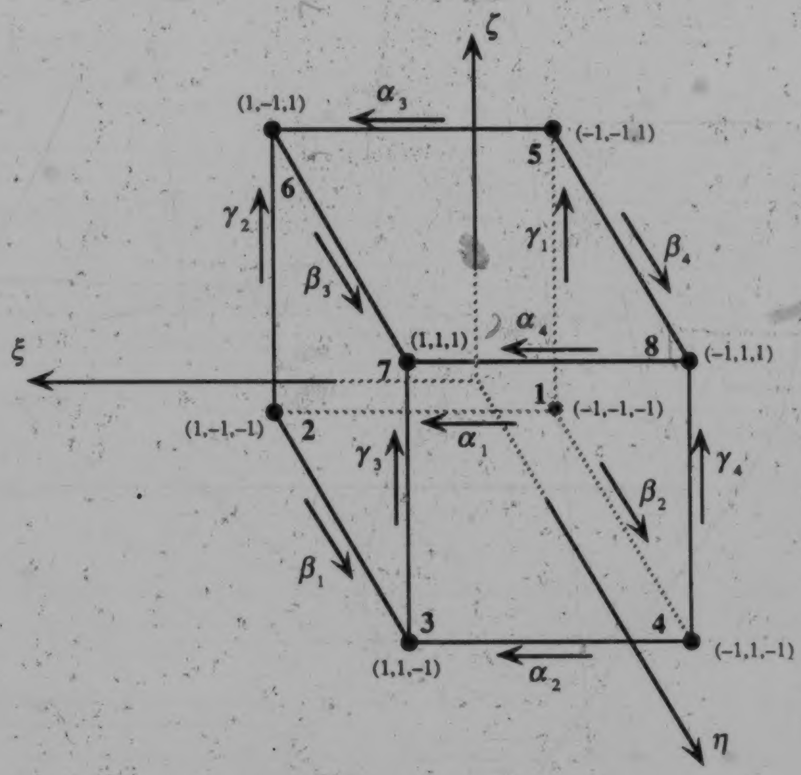

ζ
α₃
(1,-1,1)
(-1,-1,1)
5
6
γ₂
γ₁
β₄
β₃
α₄
(1,1,1)
8
(-1,1,1)
ξ
7
1
(-1,-1,-1)
(1,-1,-1)
2
α₁
γ₃
γ₄
β₂
β₁
3
4
(-1,1,-1)
(1,1,-1)
α₂
η

Figure C–6

L8 brick element with local coordinates and weighting directions.

increase in the direction of the ξ-coordinate. Thus

$$N_1^\xi = \frac{1}{2}(1-\xi);$$
$$N_2^\xi = \frac{1}{2}(1+\xi);$$

The trilinear interpolation functions are found as the product of the linear trial functions, thus

$$\begin{aligned} N_i &= N_i^\xi N_i^\eta N_i^\zeta \\ &= \frac{1}{8}(1+\xi\xi_i)(1+\eta\eta_i)(1+\zeta\zeta_i). \end{aligned} \tag{C.33}$$

where i varies from 1 to 8. The nodal values of ξ, η, and ζ are shown in Figure C–6 and are given

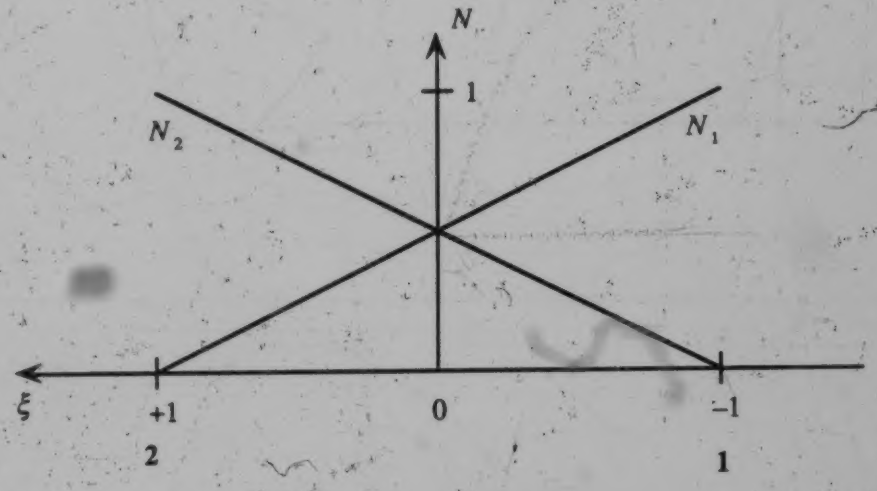


Figure C–7

Linear trial functions for ξ-coordinate.

below:

$$\left.\begin{aligned} \xi_i &= \langle -1, +1, +1, -1, -1, +1, +1, -1 \rangle; \\ \eta_i &= \langle -1, -1, +1, +1, -1, -1, +1, +1 \rangle; \\ \zeta_i &= \langle -1, -1, -1, -1, +1, +1, +1, +1 \rangle. \end{aligned}\right\} \qquad (C.34)$$

The derivatives of the trilinear interpolation functions are found to have a similar general form:

$$\left.\begin{aligned} N_{i,\xi} &\equiv \frac{\partial N_i}{\partial \xi} = \frac{\xi_i}{8}(1+\eta\eta_i)(1+\zeta\zeta_i); \\ N_{i,\eta} &\equiv \frac{\partial N_i}{\partial \eta} = \frac{\eta_i}{8}(1+\xi\xi_i)(1+\zeta\zeta_i); \\ N_{i,\zeta} &\equiv \frac{\partial N_i}{\partial \zeta} = \frac{\zeta_i}{8}(1+\xi\xi_i)(1+\eta\eta_i); \end{aligned}\right\} \qquad (C.35)$$

The L8-element asymmetric weighting functions are constructed from the two-dimensional analogue: the quadrilateral element. The weighting parameters are shown in Figure C–6. The three-dimensional functions are then constructed as products of the one-dimensional asymmetric functions, just as for the quadrilateral element. This follows the approach of Voss (1978). For example, at node 1:

$$W_1^* = G_1(\xi, \alpha_1) H_1(\eta, \beta_2) I_1(\zeta, \gamma_1). \qquad (C.36)$$

The one-dimensional asymmetric function G can be expanded as

$$\begin{aligned} G_1(\xi,\alpha_1) &= 1-\frac{x}{h}+3\alpha_1\frac{x^2}{h^2}-3\alpha_1\frac{x}{h} \\ &= 1-\frac{\xi+1}{2}+3\alpha_1\left[\frac{\xi^2+2\xi+1}{4}-\frac{\xi+1}{2}\right] \\ &= \frac{1-\xi}{2}-3\alpha_1\frac{(1-\xi)}{2}\frac{(1+\xi)}{2} \\ &= N_1^\xi-3\alpha_1N_1^\xi N_2^\xi. \end{aligned}$$

Similarly, the other two one-dimensional asymmetric functions can found:

$$\begin{aligned} H_1(\eta,\beta_2) &= N_1^\eta-3\beta_2N_1^\eta N_2^\eta; \\ I_1(\zeta,\gamma_1) &= N_1^\zeta-3\gamma_1N_1^\zeta N_2^\zeta; \end{aligned}$$

The other asymmetric weighting functions can be developed in a similar manner, leading to the complete set of eight:

$$\left.\begin{aligned} W_1^* &= (N_1^\xi-3\alpha_1N_1^\xi N_2^\xi)(N_1^\eta-3\beta_2N_1^\eta N_2^\eta)(N_1^\zeta-3\gamma_1N_1^\zeta N_2^\zeta) \\ W_2^* &= (N_2^\xi+3\alpha_1N_1^\xi N_2^\xi)(N_1^\eta-3\beta_1N_1^\eta N_2^\eta)(N_1^\zeta-3\gamma_2N_1^\zeta N_2^\zeta) \\ W_3^* &= (N_2^\xi+3\alpha_2N_1^\xi N_2^\xi)(N_2^\eta+3\beta_1N_1^\eta N_2^\eta)(N_1^\zeta-3\gamma_3N_1^\zeta N_2^\zeta) \\ W_4^* &= (N_1^\xi-3\alpha_2N_1^\xi N_2^\xi)(N_2^\eta+3\beta_2N_1^\eta N_2^\eta)(N_1^\zeta-3\gamma_4N_1^\zeta N_2^\zeta) \\ W_5^* &= (N_1^\xi-3\alpha_3N_1^\xi N_2^\xi)(N_1^\eta-3\beta_4N_1^\eta N_2^\eta)(N_2^\zeta+3\gamma_1N_1^\zeta N_2^\zeta) \\ W_6^* &= (N_2^\xi+3\alpha_3N_1^\xi N_2^\xi)(N_1^\eta-3\beta_3N_1^\eta N_2^\eta)(N_2^\zeta+3\gamma_2N_1^\zeta N_2^\zeta) \\ W_7^* &= (N_2^\xi+3\alpha_4N_1^\xi N_2^\xi)(N_2^\eta+3\beta_3N_1^\eta N_2^\eta)(N_2^\zeta+3\gamma_3N_1^\zeta N_2^\zeta) \\ W_8^* &= (N_1^\xi-3\alpha_4N_1^\xi N_2^\xi)(N_2^\eta+3\beta_4N_1^\eta N_2^\eta)(N_2^\zeta+3\gamma_4N_1^\zeta N_2^\zeta) \end{aligned}\right\} \tag{C.37}$$

The directions of the weighting parameters α, β, and γ are important. For example, the one-dimensional element and the direction from node 1 to node 2. Then a positive value of α will

a) decrease the value of W_1 (except at the nodes), and

b) increase the value of W_2 (except at the nodes).

The asymmetric weighting functions have a general equation form

$$W_i^* = \frac{1}{8}\left[(1+\xi\xi_i)+\xi_i\frac{3}{2}\alpha_i(1-\xi^2)\right]\left[(1+\eta\eta_i)+\eta_i\frac{3}{2}\beta_i(1-\eta^2)\right]\left[(1+\zeta\zeta_i)+\zeta_i\frac{3}{2}\gamma_i(1-\zeta^2)\right]. \tag{C.38}$$

where ξ_i, η_i, and ζ_i are found as before. The values of α_i, β_i, and γ_i are related to the edges of the finite element and can be deduced from Figure C–6. They are also given below.

$$\left.\begin{aligned}\alpha_i &= \langle \alpha_1, \alpha_1, \alpha_2, \alpha_2, \alpha_3, \alpha_3, \alpha_4, \alpha_4 \rangle;\\ \beta_i &= \langle \beta_2, \beta_1, \beta_1, \beta_2, \beta_4, \beta_3, \beta_3, \beta_4 \rangle;\\ \gamma_i &= \langle \gamma_1, \gamma_2, \gamma_3, \gamma_4, \gamma_1, \gamma_2, \gamma_3, \gamma_4 \rangle.\end{aligned}\right\} \tag{C.39}$$

The spatial derivatives of the upstream weighting functions must be evaluated by constraining the upstream weighting parameter values. Huyakorn and Nilkuha (1976) showed that for a satisfactory solution the upstream parameters in the directions other than that of differentiation variable must be set to zero:

$$\left.\begin{aligned}\frac{\partial W_i^*(\alpha_i, \beta_i, \gamma_i)}{\partial \xi} &= \frac{\partial W_i^*(\alpha_i, 0, 0)}{\partial \xi};\\ \frac{\partial W_i^*(\alpha_i, \beta_i, \gamma_i)}{\partial \eta} &= \frac{\partial W_i^*(0, \beta_i, 0)}{\partial \xi};\\ \frac{\partial W_i^*(\alpha_i, \beta_i, \gamma_i)}{\partial \zeta} &= \frac{\partial W_i^*(0, 0, \gamma_i)}{\partial \xi}.\end{aligned}\right\} \tag{C.40}$$

Therefore, the general form of Equation (C.38) can be used to form the derivatives of the asymmetric weighting functions:

$$\left.\begin{aligned}\frac{\partial W_i^*}{\partial \xi} &= \frac{\xi_i}{8}(1 - 3\alpha_i\xi)(1 + \eta\eta_i)(1 + \zeta\zeta_i);\\ \frac{\partial W_i^*}{\partial \eta} &= \frac{\eta_i}{8}(1 - 3\beta_i\eta)(1 + \xi\xi_i)(1 + \zeta\zeta_i);\\ \frac{\partial W_i^*}{\partial \zeta} &= \frac{\zeta_i}{8}(1 - 3\gamma_i\zeta)(1 + \xi\xi_i)(1 + \eta\eta_i);\end{aligned}\right\} \tag{C.41}$$

The only remaining task is to decide on the sign convention for the values of the upstream weighting parameters. This is done by inspection of the velocity term along the edge, which in practice means a comparison of the potentials at adjacent nodes. If the velocity is in the same direction as the defined direction of the upstream weighting parameter, then the parameter takes a positive sign. If the velocity is in the other direction, then the parameter takes a negative sign. If the velocity is zero, there is no convection locally and the weighting parameter along that edge is zero. The values chosen for the weighting parameters will influence the degree of upstream weighting, and a value of ±1 is sufficient.

C.8 *Coordinate transformation and the jacobian matrix*

Any local coordinate is transformed into the global coordinate system within the finite element by the relation

$$\left.\begin{aligned} x &= N_i x_i; \\ y &= N_i y_i; \\ z &= N_i z_i; \end{aligned}\right\} \tag{C.23}$$

where N_i is the set of *shape* functions, which, in the isoparametric formulation, is identical to the set of trial functions, and x_i, y_i, and z_i are the global coordinate values at each of the nodes i. The transformation for the set of derivatives of any function $\phi(x, y, z)$ was given in Equations (C.19) to (C.22).

The Jacobian matrix of the transformation is, in index notation,

$$[J] = \begin{bmatrix} x_{,\xi} & y_{,\xi} & z_{,\xi} \\ x_{,\eta} & y_{,\eta} & z_{,\eta} \\ x_{,\zeta} & y_{,\zeta} & z_{,\zeta} \end{bmatrix} \tag{C.42}$$

where each of the elements of the Jacobian matrix is found by differentiating the shape functions, e.g.,

$$x_{,\xi} = \frac{\partial N_i}{\partial \xi} x_i$$

This calculation has been designed to take advantage of a pipelined vector processor such as the CDC Cyber 205. The Jacobian matrix of transformation is found for all finite elements simultaneously. The solution to the set of equations

$$\begin{Bmatrix} \frac{\partial N_i^e}{\partial x} \\ \frac{\partial N_i^e}{\partial y} \\ \frac{\partial N_i^e}{\partial z} \end{Bmatrix} = \mathbf{J}^{-1} \begin{Bmatrix} \frac{\partial N_i^e}{\partial \xi} \\ \frac{\partial N_i^e}{\partial \eta} \\ \frac{\partial N_i^e}{\partial \zeta} \end{Bmatrix}. \tag{C.21}$$

is then found quickly using an efficient vector implementation of Cramer's Rule. The transformation in a given direction is thus effectively found for each element simultaneously.

C.9 *Gaussian quadrature*

Gaussian quadrature is one of the most efficient methods of numerical integration, requiring fewer points to integrate exactly a given polynomial than many other schemes. The integrands formed by the weighted residual statments with isoparametric elements are generally polynomials of indeterminate order because of the calculation of the inverse of the Jacobian of transformation. The transformation entails division by a polynomial. Thus the order of, or number of points for, Gaussian quadrature to integrate the expressions exactly is also indeterminate. Then the choice of the order of the quadrature is limited by the available computational resources.

Gaussian quadrature will integrate exactly a polynomial of order $n = 2m - 1$, where m is the order of integration, i.e., the number of quadrature points. Integration in more than one dimension is achieved by using the quadrature scheme over each of the basis vectors. Thus a 3-point rule in three dimensions requires $3 \times 3 \times 3 = 27$ quadrature points. Tables of quadrature points are provided in the literature for a given order of numerical integration in one dimension, assuming that the coordinate varies from -1 to $+1$. Various sets of integration points have also been found for specific shapes and coordinate systems, with the general intent of minimizing the number of points required to integrate exactly a polynomial of given degree. This is advantageous because the integrand, often containing many terms, must be re-evaluated at each quadrature point. Various integration rules have been provided for the L8 and L6 elements used in FESPS.

The Gauss quadrature points for one-dimensional isoparametric coordinates varying from -1 to $+1$ are taken from Abramowitz and Stegun (1965). There is also a weighting factor associated with each quadrature point, which is also supplied in the literature. All the coordinates and weighting factors are reproduced here for completeness.

C.9.1 *Integration rules for L8 brick elements*

Two integration rules are provided:

i) An order 2 scheme ($2 \times 2 \times 2 = 8$ points), which will integrate exactly a polynomial of the form $\sum c_{ijk} x^i y^j z^k$ subject to $i, j, k \leq 3$, i.e., a complete cubic plus assorted higher order terms.

ii) A 14-point rule by Irons (1971) which will integrate exactly a complete quintic polynomial: $\sum c_{ijk} x^i y^j z^k$ subject to $i + j + k \leq 5$. A 27-point rule (order 3) would be necessary otherwise to integrate a complete quintic plus assorted higher order terms $(i, j, k \leq 5)$.

The 2-point Gauss quadrature rule in one dimension is extended to three dimensions by taking the product of the quadrature point weightings. Thus the 2-point rule, for which $\pm x_i = \pm 0.577350269189626$ and the weighting is $W_i = 1.000$ (Abramowitz and Stegun, 1965), provides eight quadrature points in three dimensions, specifically, in a cube with sides of length 2 and a coordinate origin in the centre. The specific arrangment of coordinates is shown in Table C–1.

The 14-point rule devised by Irons (1971) for 8-noded brick elements is described by the

	ξ	η	ζ	W_i
1	0.577350269189626	0.577350269189626	0.577350269189626	1.000000000000000
2	−0.577350269189626	0.577350269189626	0.577350269189626	1.000000000000000
3	0.577350269189626	−0.577350269189626	0.577350269189626	1.000000000000000
4	−0.577350269189626	−0.577350269189626	0.577350269189626	1.000000000000000
5	0.577350269189626	0.577350269189626	−0.577350269189626	1.000000000000000
6	−0.577350269189626	0.577350269189626	−0.577350269189626	1.000000000000000
7	0.577350269189626	−0.577350269189626	−0.577350269189626	1.000000000000000
8	−0.577350269189626	−0.577350269189626	−0.577350269189626	1.000000000000000

Table C–1

Gauss points and weightings for 2-point rule in L8 elements.

following quadrature formula:

$$\iiint_{-1}^{+1} f(\xi,\eta,\zeta)\,d\xi\,d\eta\,d\zeta = B_6[f(-b,0,0)+f(b,0,0)+f(0,-b,0)+f(0,b,0)+f(0,0,-b)+f(0,0,b)]$$
$$+ C_8[f(-c,-c,-c)+f(c,-c,-c)+f(c,c,-c)+f(c,c,c)$$
$$+ f(-c,-c,c)+f(-c,c,-c)+f(c,-c,c)+f(-c,c,c)] \qquad \text{(C.43)}$$

where

$$\left.\begin{aligned} B_6 &= 0.886426593 \\ C_8 &= 0.335180055 \\ b &= 0.795822426 \\ c &= 0.758786911 \end{aligned}\right\} \qquad \text{(C.44)}$$

C.9.2 *Integration rules for L6 triangular prism elements*

Again, two integration rules are provided:

i) An order 2 quadrature scheme ($2\times2\times2=8$ points), which will integrate exactly a polynomial of the form $\sum c_{ijk}x^i y^j z^k$ subject to $i,j,k \le 3$, i.e., a complete cubic plus assorted higher order terms.

ii) A scheme based on a 7-point scheme for triangular elements and an order 2 scheme for one dimension. The 7-point scheme is from Cowper (1973), and, combined with the 2-point scheme

	w_1	w_2	ζ	W_i
1	0.577350269189626	0.577350269189626	0.577350269189626	1.000000000000000
2	−0.577350269189626	0.577350269189626	0.577350269189626	1.000000000000000
3	0.577350269189626	−0.577350269189626	0.577350269189626	1.000000000000000
4	−0.577350269189626	−0.577350269189626	0.577350269189626	1.000000000000000
5	0.577350269189626	0.577350269189626	−0.577350269189626	1.000000000000000
6	−0.577350269189626	0.577350269189626	−0.577350269189626	1.000000000000000
7	0.577350269189626	−0.577350269189626	−0.577350269189626	1.000000000000000
8	−0.577350269189626	−0.577350269189626	−0.577350269189626	1.000000000000000

Table C–2

Gauss points and weightings for 8-point rule in L6 elements.

along the isoparametric coordinate, forms a 14-point scheme which exactly integrates an incomplete quintic polynomial: $\sum c_{ijk}x^iy^jz^k$ subject to $i+j+k \leq 5$ and $j \leq 3$.

The quadrature formulae for the triangular prism element are found by combining the formula for the linear strain triangle and one isoparametric coordinate. The 8-point formula is found by combining the 4-point formula for a linear strain triangle (Cowper, 1973) and the 2-point formula for a single isoparametric coordinate. The combined rule can then be made as tabulated in Table C–2.

	w_1	w_2	ζ	W_i
1	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	0.225
2	α_1	β_1	β_1	0.132394152788506
3	β_1	α_1	β_1	0.132394152788506
4	β_1	β_1	α_1	0.132394152788506
5	α_2	β_2	β_2	0.125939180544827
6	β_2	α_2	β_2	0.125939180544827
7	β_2	β_2	α_2	0.125939180544827

Table C–3

Gauss points and weightings for 7-point rule in a linear triangle.

A suitable 14-point rule is found by combining the 7-point rule for a linear strain triangle

(Cowper, 1973) with the 2-point rule for the isoparametric coordinate. The coordinates and weights listed in Table C–3 are used with each of the two quadrature points for the isoparametric coordinate. The values of the coordinates are given by:

$$\left.\begin{aligned} \alpha_1 &= 0.059715871789770 \\ \beta_1 &= 0.470142064105115 \\ \alpha_2 &= 0.797426985353087 \\ \beta_2 &= 0.101286507323456 \end{aligned}\right\} \tag{C.45}$$

C.10 *Triangular Prism Elements*

The triangular prism element, or L6 element, is constructed from one isoparametric coordinate and two sets of areal coordinates for linear triangles. The linear triangle is one of the original finite elements, see, for example, Cook (1981). The element is depicted in Figure C–8, and the trial functions are given by:

$$\begin{Bmatrix} w_1 \\ w_2 \\ w_3 \end{Bmatrix} = [G]^{-1} \begin{Bmatrix} 1 \\ x \\ y \end{Bmatrix}. \tag{C.46}$$

where

$$[G] = \begin{bmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{bmatrix}.$$

w_1, w_2, and w_3 are referred to as *areal* coordinates. The coordinates are called areal because they are formed by the ratio of areas formed by drawing lines from the vertices to the point in the element, divided by the total area, A, of the triangle. Thus $w_1 = A_1/A$, etc. There are actually only two independent coordinates because $w_1 + w_2 + w_3 = 1$. A redundant coordinate can be chosen, e.g., w_3:

$$w_3 = 1 - w_1 - w_2. \tag{C.47}$$

Each of the areal coordinates w_i varies linearly from 0 all along the side opposite the vertex i, to 1 at the vertex i. Each intermediate value lies on a line parallel to the side defined by $w_i = 0$. Note that the areal coordinates are automatically isoparametric.

We construct a three-dimensional triangular prism element using the linear triangle and an isoparametric coordinate ζ. The triangular surfaces of the prism are defined by $\zeta = +1$ (nodes 1, 2,

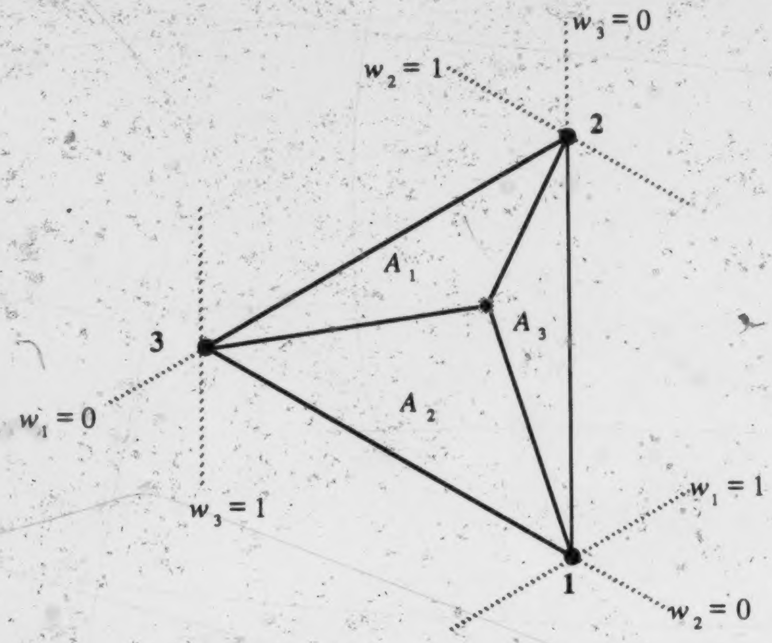


Figure C–8

The linear triangle.

and 3) and by $\zeta = -1$ (nodes 4, 5, and 6). The trial function associated with the ζ-coordinate is:

$$\left.\begin{aligned} N_i^\zeta &= \frac{1}{2}(1+\zeta) \quad i = 1,2,3; \\ N_i^\zeta &= \frac{1}{2}(1-\zeta) \quad i = 4,5,6; \end{aligned}\right\} \tag{C.48}$$

Therefore the trial functions for the three-dimensional triangular prism are formed by the product of N_i^ζ with the areal coordinates:

$$\left.\begin{aligned}
N_1 &= \frac{w_1}{2}(1+\zeta)\\
N_2 &= \frac{w_2}{2}(1+\zeta)\\
N_3 &= \frac{1-w_1-w_2}{2}(1+\zeta)\\
N_4 &= \frac{w_1}{2}(1-\zeta)\\
N_5 &= \frac{w_2}{2}(1-\zeta)\\
N_6 &= \frac{1-w_1-w_2}{2}(1-\zeta)
\end{aligned}\right\} \tag{C.49}$$

The triangular prism element is illustrated in Figure C–9.

The derivatives required are:

$$\frac{\partial N_i}{\partial w_1},\quad \frac{\partial N_i}{\partial w_2},\quad \frac{\partial N_i}{\partial \zeta}.$$

The derivatives can be formed directly from the L6 element shape functions, but there is no general form as there was for the L8 elements. Thus:

$$\left.\begin{aligned}
\frac{\partial N_1}{\partial w_1} &= \frac{1}{2}(1+\zeta);\\
\frac{\partial N_1}{\partial w_2} &= 0;\\
\frac{\partial N_1}{\partial \zeta} &= \frac{w_1}{2};
\end{aligned}\right\} \tag{C.50a}$$

$$\left.\begin{aligned}
\frac{\partial N_2}{\partial w_1} &= 0;\\
\frac{\partial N_2}{\partial w_2} &= \frac{1}{2}(1+\zeta);\\
\frac{\partial N_2}{\partial \zeta} &= \frac{w_2}{2};
\end{aligned}\right\} \tag{C.50b}$$

$$\left.\begin{aligned}
\frac{\partial N_3}{\partial w_1} &= -\frac{1}{2}(1+\zeta);\\
\frac{\partial N_3}{\partial w_2} &= -\frac{1}{2}(1+\zeta);\\
\frac{\partial N_3}{\partial \zeta} &= \frac{1-w_1-w_2}{2};
\end{aligned}\right\} \tag{C.50c}$$

$$\left.\begin{aligned}
\frac{\partial N_4}{\partial w_1} &= \frac{1}{2}(1-\zeta);\\
\frac{\partial N_4}{\partial w_2} &= 0;\\
\frac{\partial N_4}{\partial \zeta} &= -\frac{w_1}{2};
\end{aligned}\right\} \tag{C.50d}$$

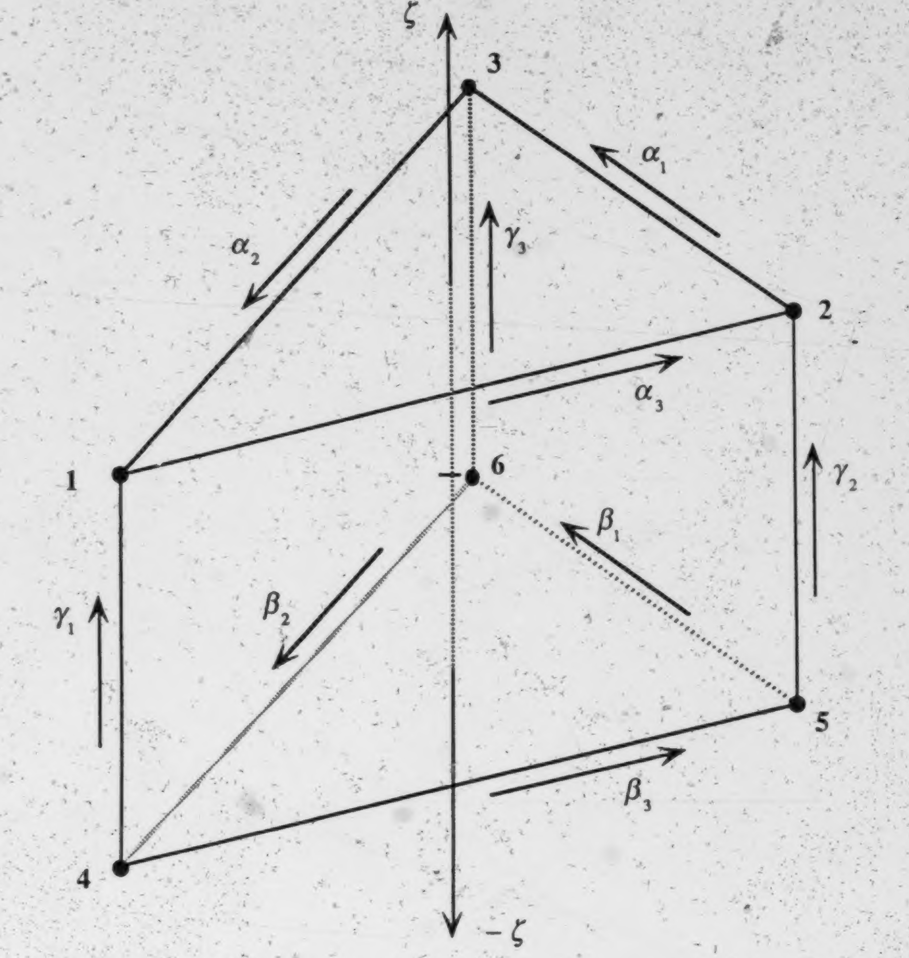

ζ
3
α₁
α₂
γ₃
2
α₃
1
6
γ₂
β₁
β₂
γ₁
5
β₃
4
– ζ

Figure C–9

L6 triangular prism element with local isoparametric coordinate and weighting directions.

$$\left.\begin{aligned}\frac{\partial N_5}{\partial w_1} &= 0;\\ \frac{\partial N_5}{\partial w_2} &= \frac{1}{2}(1-\zeta);\\ \frac{\partial N_5}{\partial \zeta} &= -\frac{w_2}{2};\end{aligned}\right\} \qquad \text{(C.50e)}$$

$$\left.\begin{aligned}\frac{\partial N_6}{\partial w_1} &= -\frac{1}{2}(1-\zeta);\\ \frac{\partial N_6}{\partial w_2} &= -\frac{1}{2}(1-\zeta);\\ \frac{\partial N_6}{\partial \zeta} &= -\frac{1-w_1-w_2}{2};\end{aligned}\right\} \tag{C.50f}$$

C.11 *Asymmetric Weighting Functions for the Triangular Prism Elements*

Asymmetric weighting functions for linear triangular elements were developed by Huyakorn (1976), and are shown in Equation (C.31).

$$\left.\begin{aligned}F_1 &= 3(\alpha_2 w_3 w_1 - \alpha_3 w_2 w_1);\\ F_2 &= 3(\alpha_3 w_1 w_2 - \alpha_1 w_3 w_2);\\ F_3 &= 3(\alpha_1 w_2 w_3 - \alpha_2 w_1 w_3).\end{aligned}\right\} \tag{C.31}$$

The weighting parameter directions were illustrated in Figure C–5. The weighting parameters on the triangular surface defined by nodes 4, 5, and 6 are found similarly:

$$\left.\begin{aligned}F_4 &= 3(\beta_2 w_3 w_1 - \beta_3 w_2 w_1);\\ F_5 &= 3(\beta_3 w_1 w_2 - \beta_1 w_3 w_2);\\ F_6 &= 3(\beta_1 w_2 w_3 - \beta_2 w_1 w_3).\end{aligned}\right\} \tag{C.51}$$

The weighting parameters β_1, β_2, and β_3 were illustrated in Figure C–9. The asymmetric weighting functions for the isoparametric coordinate are the same as for the L8 brick element, viz.:

$$\left.\begin{aligned}G_i &= \frac{1+\zeta}{2} - \frac{3}{4}\gamma_i(1-\gamma^2) \quad i = 1,2,3;\\ G_i &= \frac{1-\zeta}{2} + \frac{3}{4}\gamma_{i-3}(1-\gamma^2) \quad i = 4,5,6;\end{aligned}\right\} \tag{C.52}$$

The asymmetric weighting functions for the L6 element are then formed from the product of the asymmetric weighting functions for the linear triangle and the isoparametric coordinate. Thus

$$\left.\begin{aligned}
W_1^* &= [w_1 + 3(\alpha_2 w_3 w_1 - \alpha_3 w_2 w_1)]\left[\frac{1+\zeta}{2} - \frac{3}{4}\gamma_1(1-\gamma^2)\right];\\
W_2^* &= [w_2 + 3(\alpha_3 w_1 w_2 - \alpha_1 w_3 w_2)]\left[\frac{1+\zeta}{2} - \frac{3}{4}\gamma_2(1-\gamma^2)\right];\\
W_3^* &= [w_3 + 3(\alpha_1 w_2 w_3 - \alpha_2 w_1 w_3)]\left[\frac{1+\zeta}{2} - \frac{3}{4}\gamma_3(1-\gamma^2)\right];\\
W_4^* &= [w_1 + 3(\beta_2 w_3 w_1 - \beta_3 w_2 w_1)]\left[\frac{1-\zeta}{2} + \frac{3}{4}\gamma_1(1-\gamma^2)\right];\\
W_5^* &= [w_2 + 3(\beta_3 w_1 w_2 - \beta_1 w_3 w_2)]\left[\frac{1-\zeta}{2} + \frac{3}{4}\gamma_2(1-\gamma^2)\right];\\
W_6^* &= [w_3 + 3(\beta_1 w_2 w_3 - \beta_2 w_1 w_3)]\left[\frac{1-\zeta}{2} + \frac{3}{4}\gamma_3(1-\gamma^2)\right];
\end{aligned}\right\} \tag{C.53}$$

where $w_3 = 1 - w_1 - w_2$.

The spatial derivatives of the upstream weighting functions must be evaluated by constraining the upstream weighting parameter values, as in the case of the L8 elements. Thus we have:

$$\left.\begin{aligned}
\frac{\partial W_i^*(\alpha_i,\beta_i,\gamma_i)}{\partial \zeta} &= \frac{\partial W_i(0,0,\gamma_i)}{\partial \zeta};\\
\frac{\partial W_i^*(\alpha_i,\beta_i,\gamma_i)}{\partial w_1} &= \frac{\partial W_i(\alpha_i,0,0)}{\partial w_1};\\
\frac{\partial W_i^*(\alpha_i,\beta_i,\gamma_i)}{\partial w_2} &= \frac{\partial W_i(\alpha_i,0,0)}{\partial w_2};
\end{aligned}\right\}\quad i = 1,2,3. \tag{C.54a}$$

$$\left.\begin{aligned}
\frac{\partial W_i^*(\alpha_{i-3},\beta_{i-3},\gamma_{i-3})}{\partial \zeta} &= \frac{\partial W_i(0,0,\gamma_{i-3})}{\partial \zeta};\\
\frac{\partial W_i^*(\alpha_{i-3},\beta_{i-3},\gamma_{i-3})}{\partial w_1} &= \frac{\partial W_i(0,\beta_{i-3},0)}{\partial w_1};\\
\frac{\partial W_i^*(\alpha_{i-3},\beta_{i-3},\gamma_{i-3})}{\partial w_2} &= \frac{\partial W_i(0,\beta_{i-3},0)}{\partial w_2};
\end{aligned}\right\}\quad i = 4,5,6. \tag{C.54b}$$

Thus the derivatives can be formed directly from the asymmetric weighting functions shown in Equation (C.53). The derivatives are shown below in the ordered fashion used for vector processor calculation. Zero terms are left in to illustrate better the vector structure.

$$\left.\begin{aligned}
\frac{\partial W_1^*}{\partial w_1} &= \frac{1+\zeta}{2}[1 + 3\alpha_2(1-2w_1) - 3w_2(\alpha_2+\alpha_3)];\\
\frac{\partial W_2^*}{\partial w_1} &= \frac{1+\zeta}{2}[0 + 0\alpha_2(1-2w_1) + 3w_2(\alpha_3+\alpha_1)];\\
\frac{\partial W_3^*}{\partial w_1} &= \frac{1+\zeta}{2}[-1 - 3\alpha_2(1-2w_1) + 3w_2(\alpha_1-\alpha_2)];\\
\frac{\partial W_4^*}{\partial w_1} &= \frac{1-\zeta}{2}[1 + 3\beta_2(1-2w_1) - 3w_2(\beta_2+\beta_3)];\\
\frac{\partial W_5^*}{\partial w_1} &= \frac{1-\zeta}{2}[0 + 0\beta_2(1-2w_1) + 3w_2(\beta_3+\beta_1)];\\
\frac{\partial W_6^*}{\partial w_1} &= \frac{1-\zeta}{2}[-1 - 3\beta_2(1-2w_1) + 3w_2(\beta_1-\beta_2)];
\end{aligned}\right\} \tag{C.55}$$

$$\left.\begin{aligned}
\frac{\partial W_1^*}{\partial w_2} &= \frac{1+\zeta}{2}[0 + 0\alpha_1(1-2w_2) - 3w_1(\alpha_2+\alpha_3)];\\
\frac{\partial W_2^*}{\partial w_2} &= \frac{1+\zeta}{2}[1 - 3\alpha_1(1-2w_2) + 3w_1(\alpha_3+\alpha_1)];\\
\frac{\partial W_3^*}{\partial w_2} &= \frac{1+\zeta}{2}[-1 + 3\alpha_1(1-2w_2) - 3w_1(\alpha_1-\alpha_2)];\\
\frac{\partial W_4^*}{\partial w_2} &= \frac{1-\zeta}{2}[0 + 0\beta_1(1-2w_2) - 3w_1(\beta_2+\beta_3)];\\
\frac{\partial W_5^*}{\partial w_2} &= \frac{1-\zeta}{2}[1 - 3\beta_1(1-2w_2) + 3w_1(\beta_3+\beta_1)];\\
\frac{\partial W_6^*}{\partial w_2} &= \frac{1-\zeta}{2}[-1 + 3\beta_1(1-2w_2) - 3w_1(\beta_1-\beta_2)];
\end{aligned}\right\} \tag{C.56}$$

$$\left.\begin{aligned}
\frac{\partial W_1^*}{\partial \zeta} &= \frac{w_1}{2}(1+3\gamma_1\zeta);\\
\frac{\partial W_2^*}{\partial \zeta} &= \frac{w_2}{2}(1+3\gamma_2\zeta);\\
\frac{\partial W_3^*}{\partial \zeta} &= \frac{(1-w_1-w_2)}{2}(1+3\gamma_3\zeta);\\
\frac{\partial W_4^*}{\partial \zeta} &= \frac{w_1}{2}(-1+3\gamma_1\zeta);\\
\frac{\partial W_5^*}{\partial \zeta} &= \frac{w_2}{2}(-1+3\gamma_2\zeta);\\
\frac{\partial W_6^*}{\partial \zeta} &= \frac{(1-w_1-w_2)}{2}(-1+3\gamma_3\zeta);
\end{aligned}\right\} \tag{C.57}$$

Appendix D

DEVELOPMENT OF THE FULLY COUPLED FINITE ELEMENT PROGRAM FESPS

The development of the computer program is the third major stage in the solution of the problem defined for this study. It is also the most time-consuming. An overview of the model program is given first, with subsequent explanation of the component parts, or modules, and the major subroutines constituting the modules. Implementation of the boundary conditions, the well model, as a module, follows the overview. Program flow and algorithms are explained using flow-charts and so-called pseudocode. Flow charts graphically illustrate the sequence of instructions. Pseudocode represents the sequence of instructions as a series of instructions that might be read by a less pedantic computer language compiler than exists today. Numerical computational efficiency is an important aspect of a large computer program. The implementation of the creation of the set of linear equation coefficients and the solution of the equations is given. Space is given to the problem of renumbering the degrees of freedom to reduce the size of the coefficient matrix of the set of linear equations to be solved. Some aspects of the specific implementation of the algorithms on a vector computer, the CDC Cyber 205, are given where they differ from that of a conventional, sequential, computer.

D.1 *Overview of the model*

An overview of the program flow is provided by the chart in Figure D–1. The main program is called FESPS, an acronym for Finite Element Steam Parting Simulator. The renumbering pre-processor is called RENUM. Most of the items in the flow diagram are modules which can be tested independently with their own data. The rest of this chapter briefly describes the program flow following the flow diagram.

D.1.1 *Renumbering preprocessor*

The RENUM preprocessor, fully described in §D.9, processes the geometric connectivity data between the finite elements and the mesh nodes, and the distribution of the degrees of freedom (S_w, S_o, p, T). The degrees of freedom are numbered by RENUM to minimize the bandwidth of the coefficient matrix using the supplied connectivity data. The degree of freedom numbering is written to a file as the steering array that later will be used by FESPS to fill the Jacobian matrix for the linear equation solver. This output file is used, unchanged, as input to FESPS.

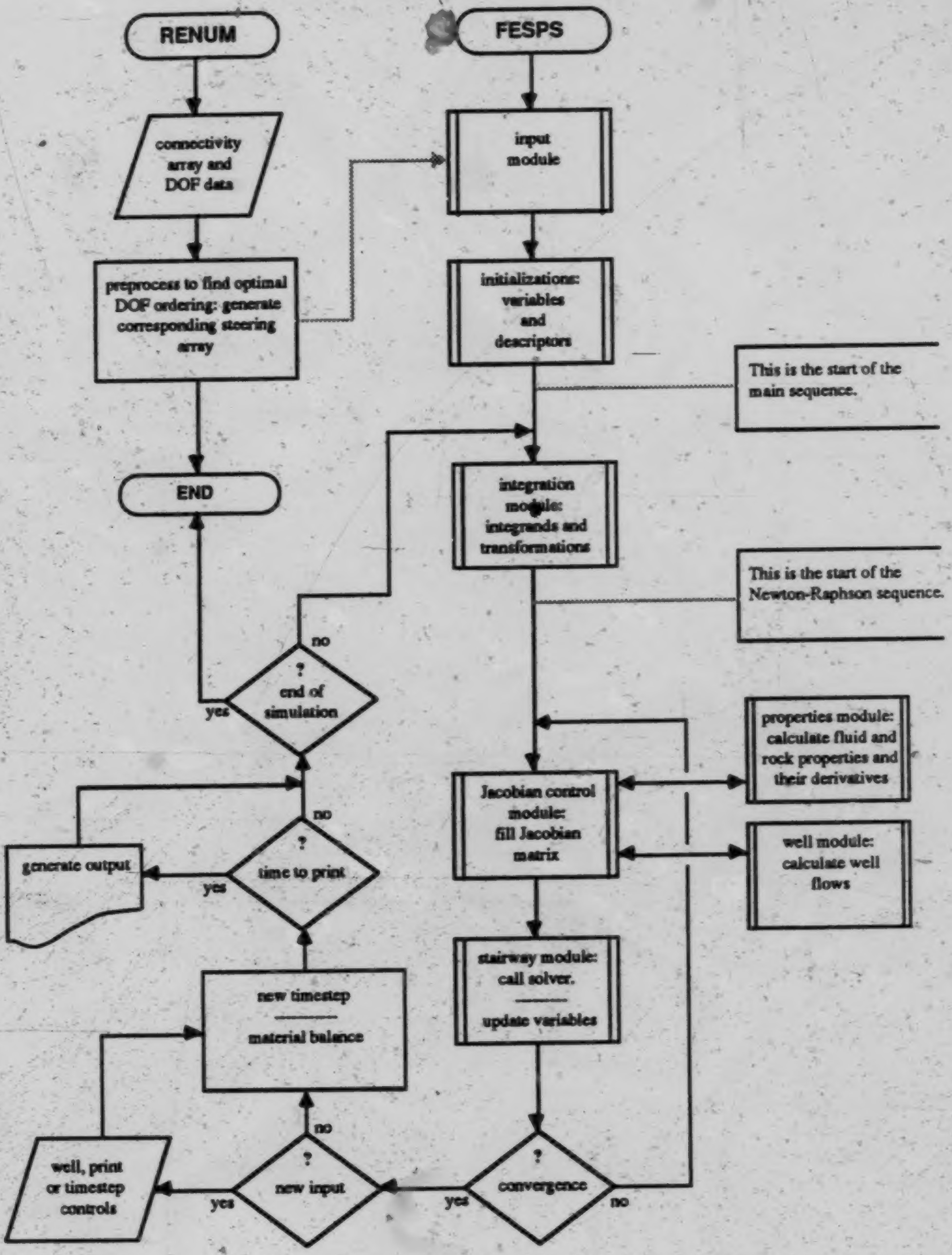

RENUM
connectivity array and DOF data
preprocess to find optimal DOF ordering: generate corresponding steering array
END
FESPS
input module
initializations: variables and descriptors
This is the start of the main sequence.
integration module: integrands and transformations
This is the start of the Newton-Raphson sequence.
properties module: calculate fluid and rock properties and their derivatives
Jacobian control module: fill Jacobian matrix
well module: calculate well flows
stairway module: call solver.
update variables
? convergence
yes
no
? new input
yes
no
well, print or timestep controls
new timestep
material balance
? time to print
yes
no
generate output
? end of simulation
yes
no

Figure D–1

Overview of FESPS program flow.

D.1.2 *I/O module*

Three files are provided as input to FESPS. Only one is supplied directly by the user. The other two are the steering array information, provided as the output file from RENUM, and the restart file, generated on a previous simulation. The restart file is unnecessary if it is not to be used. The user-supplied input file is simplified by the use of keywords and unlimited user comments. A variety of units are supported, referred to by name in the input file, including SI and variations, and common US units. The user then supplies the usual geometric data and initial values of the primary variables. Finally, printing parameters and well information are specified. Errors while the input files are read will cause the program to abort with quite specific information about where and why the error occurred.

D.1.3 *Initializations*

Two main functions are supported in this module: initialization of undefined variables and the assignment of vector descriptors. All the undefined variables are set to zero. Fluid pressures and solution-gas ratios can be found from a calculation of fluid equilibrium, if requested. Vector descriptors are used on the CDC Cyber 205 computer to refer to portions of arrays and often considerably improve the computational performance. The descriptors also simplify the program code, making it easier for someone to follow.

D.1.4 *Single time step solution*

The simulation proper starts here. All the integrals are formed at the Gauss quadrature points, which are different for each of the two element types L8 and L6. The integrals are then evaluated using the Gauss quadrature rules for the respective elements. The program now enters the Newton-Raphson iteration sequence that linearizes the problem. The derivatives are found numerically using a scheme which has been simplified as much as possible by suitable algebraic analysis. The fluid and rock properties, and certain derivatives, are found at this point to be used in forming the coefficients. Next, well flows are calculated. The coefficients are formed from this information and distributed into the stiffness (coefficient) matrix and the right-hand side constant vector. The coefficients are distributed according to the the optimal degree of freedom ordering found by RENUM.

The primary variables are shifted in turn, each time repeating the sequence outlined in the previous paragraph, incrementally forming the derivatives in the correct places in the stiffness matrix. Once the stiffness matrix has been completed the set of equations is solved using the vectorized linear equation solver STAIRWAY. The equations are now tested for convergence. The system is iterated until convergence occurs and the solution found for the current time step, or a preset limit on the number of iterations is exceeded. If the Newton-Raphson iteration fails, it is repeated with a different time step value.

Upon the successful completion of the solution to the new time level the input file will be read if the simulation time corresponds to the time at which the new information was to become active. New well information, printing controls, or time step control information may be read. The new time step is now calculated using supplied parameters and the norms of the vectors of change in the primary variables. A material balance is also evaluated using an integral form for finite elements.

Any printing or writing to a restart file is done at this point. If the simulation is not at an end, it continues for another time step.

D.2 *Description of the modularity of the program*

The model exists in various modules. The purpose was to break down the problem into portions each of which could be adequately managed by one person. Each module would be designed to be plugged into the model with little effort at the interface, while permitting the developer to test or improve the module on its own with given data. The modules are

(i) the integration routines;

(ii) the well model;

(iii) the equations and the Newton-Raphson method;

(iv) the vectorized linear solver, DSTAIR;

(v) the input/output module;

(vi) the main program with initialization code and service routines;

(vii) the node renumbering preprocessor, RENUM.

The following sections cover each module in some detail. The associated flow charts use ANSI symbols, with a bold outline to those symbols which are expanded in subsequent diagrams.

D.3 *Integration module*

This module finds the necessary integrands for each finite element, evaluated at each Gauss quadrature point. The necessary integral for each element is then found by summing the weighted

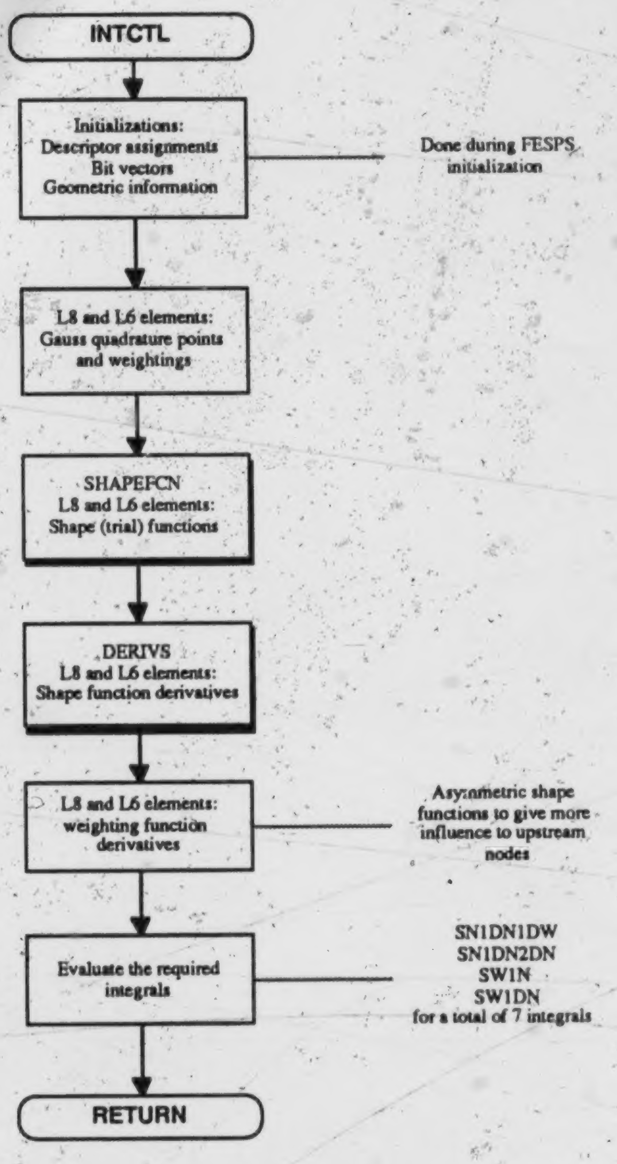

INTCTL
Initializations:
Descriptor assignments
Bit vectors
Geometric information
Done during FESPS initialization
L8 and L6 elements:
Gauss quadrature points and weightings
SHAPEFCN
L8 and L6 elements:
Shape (trial) functions
DERIVS
L8 and L6 elements:
Shape function derivatives
L8 and L6 elements:
weighting function derivatives
Asymmetric shape functions to give more influence to upstream nodes
Evaluate the required integrals
SN1DN1DW
SN1DN2DN
SW1N
SW1DN
for a total of 7 integrals
RETURN

Figure D–2

Overview of integration control subroutine.

integrands over the quadrature points.

D.3.1 *INTCTL*

The flowchart in Figure D–2 illustrates the formation of the integration terms in the formulation. The Gauss quadrature points for numerical integration must first be found for each of the L8 and L6 element types. Then the shape or trial functions and their derivatives must be found for each element, followed by evaluation of the asymmetric weighting functions and their derivatives. Once this information is available, each of the integrals can be calculated using Gauss-Legendre quadrature, in the forms required to construct the coefficient matrix.

D.3.2 *SHAPEFCN*

The calculation of the shape functions is quite simple because they are independent of position in the global coordinate system. This is shown in Figure D–3. They need only be calculated for each type of element. The integration module is, however, highly vectorized. Thus the shape functions must be presented as vectors for each Gauss point containing the function values repeated for each element. As more than one kind of element is present, the function values are ordered according to the element numbering in the mesh. The final vector is formed using a pseudo-bitmask with vectors of shape functions from each of the element types.

D.3.3 *DERIVS*

The derivatives of the shape functions with respect to the local (element) coordinate system are found much the same way as the shape functions themselves. However, the ∇ terms required for the integrals are derivatives with respect to the global coordinate system. Thus the shape function derivatives must also be transformed into derivatives with respect to the global coordinates. The procedure is outlined in Figure D–4.

D.3.4 *TRANSFRM*

The transformation subroutines convert derivatives with respect to a local coordinate system into derivatives with respect to the global coordinate system. This is made quite simply using the inverse Jacobian of transformation. The code has been highly vectorized and uses a form of Cramer's rule for this purpose. The problem to be solved is:

$$\begin{Bmatrix} \phi_{,x} \\ \phi_{,y} \\ \phi_{,z} \end{Bmatrix} = [\Gamma] \begin{Bmatrix} \phi_{,\xi} \\ \phi_{,\eta} \\ \phi_{,\zeta} \end{Bmatrix}.$$

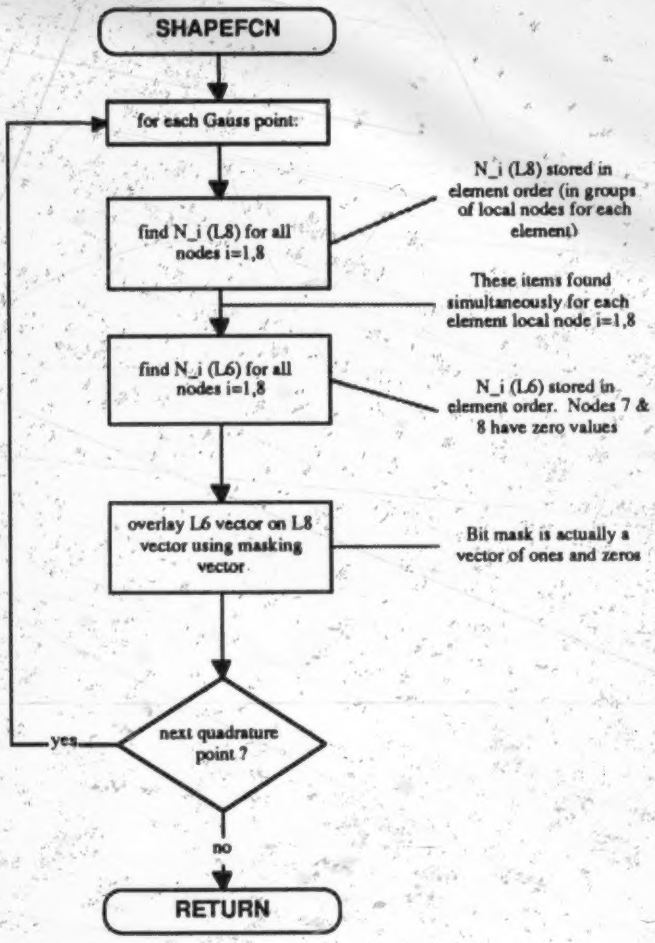

SHAPEFCN
for each Gauss point:
find N_i (L8) for all nodes i=1,8
N_i (L8) stored in element order (in groups of local nodes for each element)
These items found simultaneously for each element local node i=1,8
find N_i (L6) for all nodes i=1,8
N_i (L6) stored in element order. Nodes 7 & 8 have zero values
overlay L6 vector on L8 vector using masking vector
Bit mask is actually a vector of ones and zeros
next quadrature point ?
yes
no
RETURN

Figure D–3

Overview of calculation of shape functions.

where $[\Gamma]$ is the inverse of the Jacobian of transformation $[J]$, i.e.

$$[J] = \begin{bmatrix} x_{,\xi} & y_{,\xi} & z_{,\xi} \\ x_{,\eta} & y_{,\eta} & z_{,\eta} \\ x_{,\zeta} & y_{,\zeta} & z_{,\zeta} \end{bmatrix}.$$

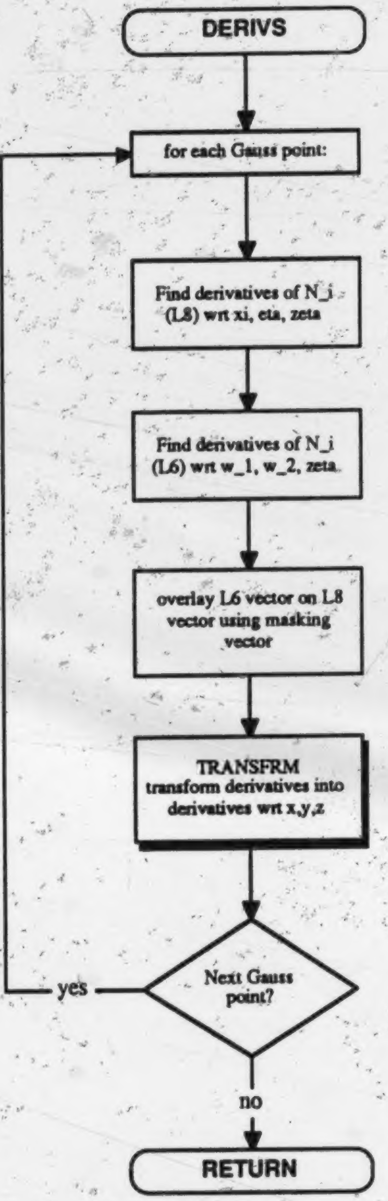

DERIVS
for each Gauss point:
Find derivatives of N_i (L8) wrt xi, eta, zeta
Find derivatives of N_i (L6) wrt w_1, w_2, zeta
overlay L6 vector on L8 vector using masking vector
TRANSFRM transform derivatives into derivatives wrt x,y,z
Next Gauss point?
yes
no
RETURN

Figure D–4

Overview of calculation of shape function derivatives.

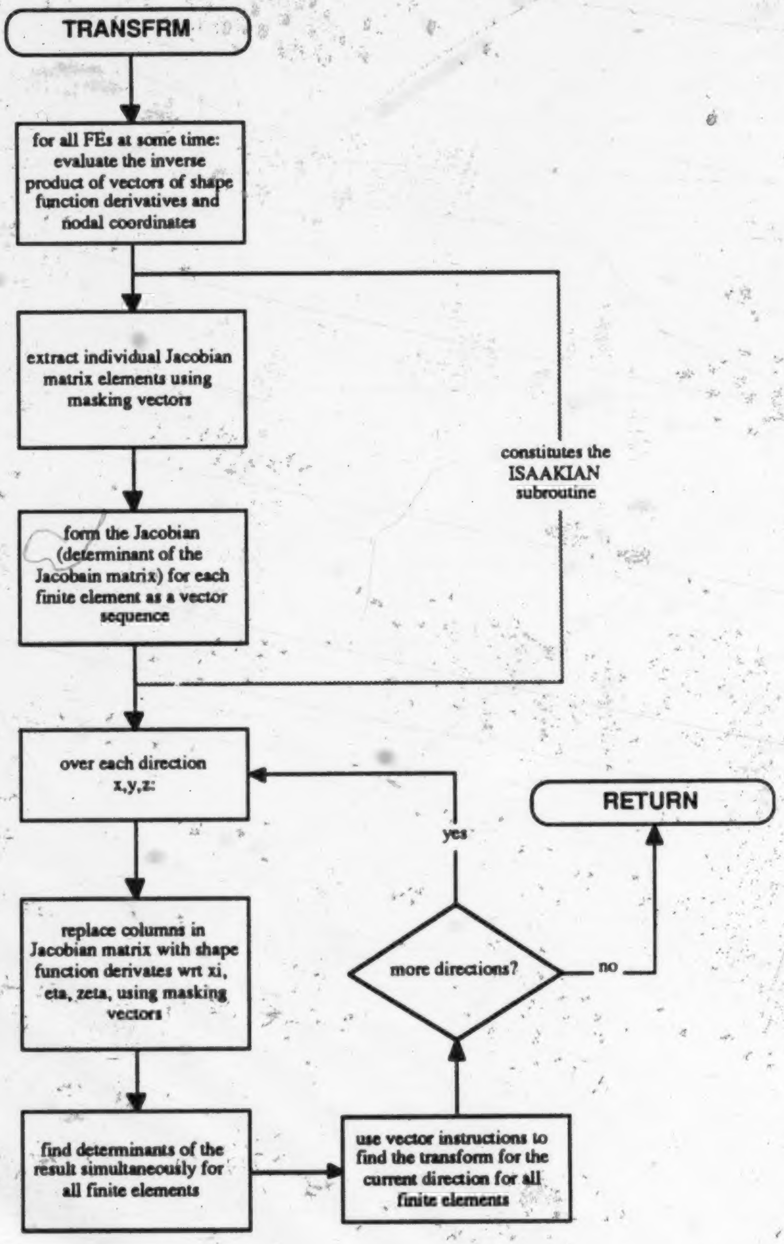

TRANSFRM
for all FEs at some time: evaluate the inverse product of vectors of shape function derivatives and nodal coordinates
extract individual Jacobian matrix elements using masking vectors
form the Jacobian (determinant of the Jacobain matrix) for each finite element as a vector sequence
constitutes the ISAAKIAN subroutine
over each direction x,y,z:
replace columns in Jacobian matrix with shape function derivates wrt xi, eta, zeta, using masking vectors
find determinants of the result simultaneously for all finite elements
use vector instructions to find the transform for the current direction for all finite elements
more directions?
yes
no
RETURN

Figure D–5

Overview of calculation of coordinate transformations.

Note the use of Einstein notation for derivatives, where

$$x_{,\xi} \equiv \frac{\partial x}{\partial \xi}.$$

and likewise with the other terms. The Jacobian of transformation must be found for every element. This procedure is illustrated in Figure D–5.

D.4 *The well model*

The well model provides values for certain integrals (fluxes) appearing in the general equation term F_i^{α}. The Jacobian terms are found by element for all elements at once. This requires that the fluxes be calculated and partitioned between nodes and elements. Thus nodes shared by two or more elements have flux contributions from each element. Every well is considered to exist along the edges of elements. Each edge is referred to as a line segment and is defined by the nodes at each end numbered in the global sense. The length of each line segment can be calculated from the coordinates of the defining nodes. The line segment is considered to be an independent entity and to have elements associated with it—in reality, the elements which share the line segment as an edge. No more than four brick elements are likely to share an edge, but there will be an additional element for each brick element replaced by two triangular prism elements. Four support nodes are defined for each element associated with a line segment. These nodes are used to supply information about pressures around the well. The support nodes comprise two pairs, each pair being the two neighbouring nodes to a node defining the line segment, numbered in the global sense, within the associated element and not on the line segment. Each associated element is also ascribed a fractional constant which describes the effective portion of radial drainage provided by the element as a fraction of a circle. This constant can also be used to increase or decrease the influence of the element on the flow to or from the specified line segment, e.g., set it to zero to remove the element as a producing or injection horizon. The well, then, is defined by its line segments which have associated elements and support nodes.

The well type indicates how the well should perform. Each well is initially assigned a given type indicated by a number:

1. pressure-specified total fluid production well;
2. bitumen rate-specified production well;
4. water rate-specified production well;

5. water and gas pressure-specified injection well;
6. water and gas rate-specified injection well;
9. shut-in well.

Types 3, 7, 8, and 10 are currently unused. A specified pressure or rate is required for the well, according to its type, and a wellbore radius. An injection well additionally requires information about the mass fraction of water—liquid and steam—in the injection stream, and the steam quality. A gas or vapour rate-specified production well has not been included.

Wells normally operate subject to a constraint. The model provides the following types of constraint:

1. maximum bitumen well rate;
2. maximum gas well rate;
3. maximum water well rate;
4. maximum well GOR;
5. maximum well WOR;
6. maximum pressure at a node (p_w);
7. minimum pressure at a node (p_w).

If a constraint is violated, that constraint temporarily redefines the well type. Violations of the GOR and WOR constraints cause the well to be shut in, while the other constraints force the well to take on that type, e.g., if the maximum water rate was exceeded, the constraint value would define the rate for a water rate-specified well type. The constraint that a production well cannot take fluids and that an injection well cannot produce fluids is automatically enforced.

The contributions to production or injection from each element are stored according to the well, line segment, and associated element. The well routines are called by `JACOBCTL`, and the contributions are brought into the set of linear equations by a call to the routine `GETFIH`. The element contributions are distributed equally between the two nodes defining the line segment.

D.4.1 *Implementation of rate-specified wells*

Rate-specified wells are easiest to implement because of the finite element formulation. The F_i^α term requires the mass flux integrated over the boundary, which is equivalent to the contribution to the well rate from a given element. The productivity index for steady-state flow in each element associated with each line segment and the sum of the indices are found first. At the same time, the nodal pressures on the line segment are checked for compliance with the maximum and minimum pressure constraints. If the pressure constraints are violated the well is converted into a pressure-specified one. Otherwise, the rate for each phase is apportioned among the elements associated with each line segment according to the productivity index of each element. The rates are stored for summation into the well rate, and also converted to mass rates for the finite element equation. Only one phase rate was specified: the other phase rates must be found. The gas rate is calculated from the gas in solution in the bitumen and the gas portion of the vapour phase. The total enthalpy and the combined enthalpies of the liquid water and steam are found for each associated element. The water and steam enthalpies are required to find the produced steam quality in the well. The mass rates of steam and liquid water are combined into one and represented by the equivalent volume rate of liquid water. Once the well rate for each component has been established, they can be compared to the constraints to check for violations. If there has been a violation, the well calculation is repeated using the appropriate well type. This might occur several times until all the constraints are satisfied.

The rate-specified injection well differs from the production well in several respects other than the sign of the flow rate. The only constraints are on pressure. The productivity index is found using the average, total, mobility. The specified rate includes the cold water equivalent of steam, and is partitioned according to the specified mass fraction and steam quality.

D.4.2 *Implementation of pressure-specified wells*

Pressure-specified well types calculate the flow rate contributions from each element using the steady-state radial flow equation. The pressure in an associated element at radius r from each line segment is found from the average of the pressures at, and the distances to, the four support nodes in that element. The mass rates for each component of flow can then be found for the finite element equation, and the equivalent volumetric rates found for the well. The well rate for each component can be checked against the specified constraints, and a new calculation made if any constraint is violated.

The choice of steady-state or pseudo-steady-state conditions to represent flow in the element to a wellbore is arbitrary here, but really is a function of the recovery process. A cyclic stimulation project would be better represented by pseudo-steady-state conditions, especially in the blowdown period, whereas a continuous injection process would be better represented by steady-state conditions.

D.5 *Outline of well calculations*

The well calculations start in the routine **THEWELLS**. This routine cycles through all the possible wells checking for active ones. If a well is active, all the injection or production information about that well is generated in the following sequence. Note that all elements refer to those associated with a line segment of a well. In the pseudocode algebra, the C-language binary operator $a += b$ is used, and has the literal meaning "increment a by amount b". This is normally rendered in FORTRAN by the statement $a = a + b$. Similarly, C-style comments are used, which are enclosed in the symbols `/* */`.

The general sequence of calculation for any one well is as follows.

```
find all the formation volume factors and enthalpies at the line segment nodes;
set all the well rates to zero;
set all the element contributions to zero;
find the partial productivity indices for all elements
                                          associated with every line segment:

                         $J^\circ = \frac{2\pi khf}{\ln {}^{r}/_{r_w} - 0.5}$
                         /* $k = (k_x + k_y)/2$; $f$ = fraction */
                         /* $r$ = average distance to all four support nodes */
start injection or production calculation depending on type of well;
store nodal contributions per element.
```

The sequence of the well rate calculation depends on the type of well:

- a pressure-specified production well;
- a rate-specified production well;
- a pressure-specified injection well;
- a rate-specified injection well;

Each case will now be discussed.

D.5.1 *Pressure-specified production well*

The calculation sequence is as follows.

```
for each line segment
        average the following properties over the pair of defining nodes:
                $\lambda_o$, $\rho_w\lambda_w$, $\rho_v\lambda_v$, $\bar{k}$, $h$, $\rho_g$, $\rho_o$, $m_g$, $m_s$;
                        /* note:  mass fraction of gas in vapour $m_g = (M_g y_g)/M_v$ */
                        /* note:  mass fraction of steam in vapour $m_s = (M_w y_s)/M_v$ */

        for each associated element
                find the contribution to the flow rate from

                        $q_i = \lambda_i J^\circ (p_w - p) \qquad i = \begin{cases} \text{oil(bitumen + gas);} \\ \text{water(liquid);} \\ \text{vapour(steam + gas).} \end{cases}$

                        /* note:  $p_w$ is the wellbore radius */
                        /* note:  $p$ is the average $p$ from the support nodes */
                        /* note:  $r$ is the average $r$ to the support nodes */

        /* calculate:  */
                oil volumetric rate $q_o = \lambda_o J^\circ$;
                water mass rate $\dot{m}_w = \rho_w \lambda_w J^\circ$;
                vapour mass rate $\dot{m}_v = \rho_v \lambda_v J^\circ$;
                steam mass rate $\dot{m}_s = m_s \dot{m}_v$;
                gas mass rate in vapour $\dot{m}_{g_v} = m_g \dot{m}_v$;
                surface oil (bitumen) rate $Q_b = q_o / B_o$;
                bitumen rate $q_b = Q_b B_b$;
                gas mass rate in solution in oil $\dot{m}_{g_o} = q_o \rho_o - q_b \rho_b$;
                total gas mass rate $\dot{m}_g = \dot{m}_{g_v} + \dot{m}_{g_o}$;
                add element contribution to well oil rate $wqo \mathrel{+}= Q_b$;
                add element contribution to well gas rate $wqg \mathrel{+}= \dot{m}_g / \rho_{g_{sc}}$;
                add element contribution to well water (liquid + steam) rate
                        $wqw \mathrel{+}= (\dot{m}_w + \dot{m}_s) / \rho_{w_{sc}}$;
                enthalpy production from water $\dot{h}_{ws} = \dot{m}_s h_s + \dot{m}_w h_w$;
                total enthalpy production from produced fluids $\dot{h}_T = \dot{m}_b h_b + \dot{m}_g h_g + \dot{h}_{ws}$;
                add element contribution to well water enthalpy rate $wqwe \mathrel{+}= \dot{h}_{ws}$;
                add element contribution to well total enthalpy rate $wqe \mathrel{+}= \dot{h}_T$;
                combine water and steam flow rates $\dot{m}_w = \dot{m}_w + \dot{m}_s$;
        end loop on associated elements;
end loop on line segments defining the well;

check for violations of the following constraints:
        fluid injection, GOR, WOR, $wqw$, $wqg$ and $wqo$;
```

D.5.2 *Rate-specified production well (oil)*

There is a slight difference in the cases of oil and water-rate specified wells. Thus the separate entries. All properties cited are the averages of the two nodes defining a given line segment. The calculation sequence is as follows.

```
set the total well oil (bitumen) rate $wqo$ = -specified rate $rate$;
                                /* negative values indicate production */
check all pressures on line segments for violation of the pressure constraints;
set $J_T = 0$;
for each line segment
        for each associated element
                increment $J_T \mathrel{+}= J^\circ \lambda_o / b_o$;
                                /* find the total productivity index of the well */

for each line segment
        for each associated element
                prorate the production according to the productivity index

                                $$Q_b = -rate \times J^\circ \frac{\lambda_o}{B_o} \frac{1}{J_T}$$

        /* calculate: */
                oil volumetric rate $q_o = Q_b B_o$;
                water mass rate $\dot{m}_w = Q_b (B_o / \lambda_o) \lambda_w \rho_w$;
                vapour molar rate $\dot{n}_v = Q_b (B_o / \lambda_o) \lambda_v (\rho_v / M_v)$;
                steam mass rate $\dot{m}_s = M_w y_s \dot{n}_v$;
                gas mass rate in vapour $\dot{m}_{g_v} = M_g y_g \dot{n}_v$;
                oil mass rate $\dot{m}_o = q_o \rho_o$;
                bitumen mass rate $\dot{m}_b = Q_b \rho_{b_{sc}}$;
                total gas mass rate $\dot{m}_g = \dot{m}_{g_v} + (\dot{m}_o - \dot{m}_b)$;
                add element contribution to well gas rate $wqg \mathrel{+}= \dot{m}_g / \rho_{g_{sc}}$;
                add element contribution to well water (liquid + steam) rate
                                $wqw \mathrel{+}= (\dot{m}_w + \dot{m}_s) / \rho_{w_{sc}}$;
                enthalpy production from water $\dot{h}_{ws} = \dot{m}_s h_s + \dot{m}_w h_w$;
                total enthalpy production from produced fluids $\dot{h}_T = \dot{m}_b h_b + \dot{m}_g h_g + \dot{h}_{ws}$;
                add element contribution to well water enthalpy rate $wqwe \mathrel{+}= \dot{h}_{ws}$;
                add element contribution to well total enthalpy rate $wqe \mathrel{+}= \dot{h}_T$;
                combine water and steam flow rates $\dot{m}_w = \dot{m}_w + \dot{m}_s$;
        end loop on associated elements;
end loop on line segments defining the well;

check for violations of the following constraints:
        GOR, WOR, $wqw$;
```

D.5.3 *Rate-specified production well (water)*

This type of well specifies the amount of liquid water produced, but *wqw* includes steam measured as the cold water equivalent (cwe). All properties cited are the averages of the two nodes defining a given line segment. The calculation sequence is as follows.

```
check all pressures on line segments for violation of the pressure constraints;
set $J_T = 0$;
for each line segment
        for each associated element
                increment $J_T \mathrel{+}= J^\circ \lambda_w / b_w$;
                                /* find the total productivity index of the well */

for each line segment
        for each associated element
                prorate the liquid water production according to the productivity index

                        $Q_w = -rate \times J^\circ \dfrac{\lambda_w}{B_w} \dfrac{1}{J_T}$
```

```
    /* calculate:  */
        oil volumetric rate $q_o = Q_w(B_w/\lambda_w)\lambda_o$;
        vapour molar rate $\dot{n}_v = Q_w(B_w/\lambda_w)\lambda_v(\rho_v/M_v)$;
        surface bitumen rate $Q_b = q_o/B_o$;
        steam mass rate $\dot{m}_s = M_w y_s \dot{n}_v$;
        gas mass rate in vapour $\dot{m}_{g_v} = M_g y_g \dot{n}_v$;
        liquid water mass rate $\dot{m}_w = Q_w \rho_{w_{sc}}$;
        add element contribution to well oil (bitumen) rate $wqo \mathrel{+}= Q_b$;
        bitumen mass rate $\dot{m}_b = Q_b \rho_{b_{sc}}$;
        oil mass rate $\dot{m}_o = q_o \rho_o$;
        total gas mass rate $\dot{m}_g = \dot{m}_{g_v} + (\dot{m}_o - \dot{m}_b)$;
        add element contribution to well gas rate $wqg \mathrel{+}= \dot{m}_g/\rho_{g_{sc}}$;
        enthalpy production from water $\dot{h}_{ws} = \dot{m}_s h_s + \dot{m}_w h_w$;
        total enthalpy production from produced fluids $\dot{h}_T = \dot{m}_b h_b + \dot{m}_g h_g + \dot{h}_{ws}$;
        add element contribution to well water enthalpy rate $wqwe \mathrel{+}= \dot{h}_{ws}$;
        add element contribution to well total enthalpy rate $wqe \mathrel{+}= \dot{h}_T$;
        combine water and steam flow rates $\dot{m}_w = \dot{m}_w + \dot{m}_s$;
        add element contribution to well water (liquid + steam) rate
                $wqw \mathrel{+}= (\dot{m}_w + \dot{m}_s)/\rho_{w_{sc}}$;
                /* specified rate was only for liquid water */
    end loop on associated elements;
end loop on line segments defining the well;

check for violations of the following constraints:
    GOR, WOR, $wqo$;
```

D.5.4 *Pressure-specified injection well*

Injection wells have the following parameters specified additionally: m_w, the mass fraction of water (steam plus liquid) in the feed, and x_{ws}, the steam quality. Note the following relationships:

$$\rho_{m_v} = \frac{\text{moles of steam + gas}}{\text{volume of gas + volume of steam}} = \frac{y_s + y_g}{\frac{y_s}{\rho_{m_s}} + \frac{y_g}{\rho_{m_g}}}$$

$$\frac{y_s}{\rho_{m_s}} + \frac{y_g}{\rho_{m_g}} = \frac{1}{\rho_{m_v}}$$

$$\Rightarrow y_s = \left[\frac{1}{\rho_{m_v}} - \frac{y_g}{\rho_{m_g}}\right]\rho_{m_s}$$

$$\Rightarrow \rho_{m_s} = \frac{y_w \rho_{m_v}}{1 - \frac{y_g \rho_{m_v}}{\rho_{m_g}}} \tag{D.1}$$

The total flow rate q_T is related to the total molar rate $\dot{n}_{wsg}$ by

$$\begin{aligned} q_T &= q_w + q_s + q_g \\ &= \dot{n}_{wsg}\left[\frac{n_w}{\rho_{m_w}} + \frac{n_s}{\rho_{m_s}} + \frac{n_g}{\rho_{m_g}}\right] \end{aligned}$$

The calculation sequence is as follows.

```
/* calculate:  */

                        $n_w = \frac{m_w/M_w}{\frac{m_w}{M-W} + \frac{1-m_w}{M_g}}$;
                        /* number of moles of water + steam */

     moles of injected gas $n_g = 1 - n_w$;

                        $y_g = \frac{n_g}{n_w x_{ws} + n_g}$;
                        /* gas as mole fraction of vapour */
     steam as mole fraction of vapour $y_s = 1 - y_g$;
     apparent molar mass of vapour $M_v = M_w y_s + M_g y_g$;

for each line segment
     find the average the following properties over the pair of defining nodes:
          total average mobility $\lambda_t = \lambda_o + \lambda_w + \lambda_g$;
          molar density of water $\rho_{m_w} = \rho_w/M_w$;
          molar density of gas $\rho_{m_g} = \rho_g/M_g$;
          molar density of vapour $\rho_{m_v} = \rho_v/M_v$;
          molar density of steam---see Equation (0.1);
          volume of the mole fraction of water $v_{m_w} = n_w(1 - x_{ws})/\rho_{m_w}$;
          volume of the mole fraction of steam $v_{m_s} = n_w x_{ws}/\rho_{m_w}$;
          volume of the mole fraction of gas $v_{m_g} = n_g/\rho_{m_g}$;

     for each associated element
```

```
        find the total flow rate

                $$q_T = \frac{\lambda_T 2\pi h f(p_w - p)}{\ln {}^{r}/r_w - 0.5)}$$
                        /* a total possible rate using the total mobility */
        total molar rate $\dot{n}_T = q_T/(v_{m_w} + v_{m_s} + v_{m_g})$;
        liquid water molar rate $\dot{n}_w = \dot{n}_T v_{m_w} \rho_{m_w}$;
        steam molar rate $\dot{n}_s = \dot{n}_T v_{m_s} \rho_{m_s}$;
        total gas molar rate $\dot{n}_g = \dot{n}_T v_{m_g} \rho_{m_g}$;
        liquid water mass rate $\dot{m}_w = \dot{n}_w M_w$;
        steam mass rate $\dot{m}_s = \dot{n}_s M_s$;
        total gas mass rate $\dot{m}_g = \dot{n}_g M_g$;
        add element contribution to well water (liquid + steam) rate
                $wqw \mathrel{+}= (\dot{m}_w + \dot{m}_s)/\rho_{w_{sc}}$;
        add element contribution to well gas rate $wqg \mathrel{+}= \dot{m}_g/\rho_{g_{sc}}$;
        enthalpy injection from water $\dot{h}_{ws} = \dot{m}_s h_s + \dot{m}_w h_w$;
        total enthalpy production from produced fluids $\dot{h}_T = \dot{m}_g h_g + \dot{h}_{ws}$;
        add element contribution to well water enthalpy rate $wqwe \mathrel{+}= \dot{h}_{ws}$;
        add element contribution to well total enthalpy rate $wqe \mathrel{+}= \dot{h}_T$;
        combine water and steam flow rates $\dot{m}_w = \dot{m}_w + \dot{m}_s$;
    end loop on associated elements;
end loop on line segments defining the well;

check for violations of the following constraints:
    production, $wqw$;
```

D.5.5 *Rate-specified injection well*

This type of well specifies the amount of water injected, as a liquid and as steam measured as the cold water equivalent (cwe). All properties cited are the averages of the two nodes defining a given line segment. The calculation sequence is as follows.

```
check all pressures on line segments for violation of the pressure constraints;
set $J_T = 0$;
for each line segment
```

```
          for each associated element
                    increment $J_T \mathrel{+}= J^\circ[(\lambda_o/b_o) + (\lambda_w/b_w) + (\lambda_v/b_v)]$;
                                        /* find the total productivity index of the well */

          /* calculate:  */

                              $n_{ws} = \frac{m_{ws}/M_w}{\frac{m_{ws}}{M_w W} + \frac{1-m_{ws}}{M_g}}$;

                                        /* moles of injected water + steam */
          moles of injected gas $n_g = 1 - n_w$;
          moles of injected liquid water $n_w = n_{ws}(1 - x_{ws})$;
          moles of injected steam $n_s = n_{ws} x_{ws}$;
                                        /* gas as mole fraction of vapour */
          steam as mole fraction of vapour $y_s = 1 - y_g$;
          apparent molar mass of vapour $M_v = M_w y_s + M_g y_g$;

for each line segment
          for each associated element
                    set the oil rate to zero $\dot{m}_o = 0$;
                    find the total molar rate:

                              $\dot{n}_T = \frac{1}{n_{ws}} \frac{\rho_{w_{sc}}}{M_w} rate \times \frac{\left[\frac{\lambda_o}{b_o} + \frac{\lambda_w}{b_w} + \frac{\lambda_v}{b_v}\right] J^\circ}{J_T}$

          /* calculate:  */
                    liquid water molar rate $\dot{n}_w = \dot{n}_T n_w$;
                    steam molar rate $\dot{n}_s = \dot{n}_T n_s$;
                    gas molar rate $\dot{n}_g = \dot{n}_T n_g$;
                    liquid water mass rate $\dot{m}_w = \dot{n}_w M_w$;
                    steam mass rate $\dot{m}_s = \dot{n}_s M_s$;
                    total gas mass rate $\dot{m}_g = \dot{n}_g M_g$;
                    add element contribution to well water (liquid + steam) rate
                              $wqw \mathrel{+}= (\dot{m}_w + \dot{m}_s)/\rho_{w_{sc}}$;
                    add element contribution to well gas rate $wqg \mathrel{+}= \dot{m}_g/\rho_{g_{sc}}$;
                    enthalpy injection from water $h_{ws} = \dot{m}_s h_s + \dot{m}_w h_w$;
```

```
        total enthalpy production from produced fluids $\dot{h}_T = \dot{m}_g h_g + \dot{h}_{ws}$;
        add element contribution to well water enthalpy rate $wqwe$ += $\dot{h}_{ws}$;
        add element contribution to well total enthalpy rate $wqe$ += $\dot{h}_T$;
        combine water and steam flow rates $\dot{m}_w = \dot{m}_w + \dot{m}_s$;
    end loop on associated elements;
end loop on line segments defining the well;

/* no other constraint violations to check for */
```

D.6 *Forming Jacobian derivatives: implementation*

It was shown in §4.9.2 that a typical derivative is given by

$$\frac{\partial S_{IJ}^{\alpha\beta(n+1)}}{\partial \psi_H^{\gamma}} = \frac{\partial(a_K)}{\partial \psi_H^{\gamma}} \delta_{KH} \Upsilon_{KIJ}. \tag{D.2}$$

Note that this derivative is a three-dimensional matrix because there are three subscripts. The Kronecker delta signifies that only one value of K need be used in the contraction because all terms for which $K \neq H$ are multiplied by zero. To avoid storing this large data structure, one notes that the derivative appears in the Jacobian as part of an inner product with the vector $\psi_J^{\beta\prime} = (1-\theta)\psi_J^{\beta n} + \theta\psi_J^{\beta(n+1)}$, resulting in the term

$$\frac{\partial(a_K)}{\partial \psi_H^{\gamma}} \delta_{KH} \Upsilon_{KIJ} \psi_J^{\beta\prime}. \tag{D.3}$$

The result of this expression is actually a two-dimensional matrix. Each column in the matrix represents one value of H, while each row represents one value of I. Thus the derivative can be built for each value of H in turn by forming the inner product with the right-hand side term in equation (D.2), and storing it in column H of the data structure holding Equation (D.3).

The formation of terms containing derivatives of $D_{IJ}^{\alpha\beta(n+1)}$ is identical to the previous example. Derivatives of $F_I^{\alpha(n+1)}$ do not occur in inner products, but have one less dimension than the previous two examples. Therefore they can also be found by entering terms into the Hth column.

D.7 *Forming the Jacobian matrix: pseudocode*

The analyses of the previous section have shown that the function derivative $\partial \mathcal{M}_I^{\alpha\prime}/\partial \psi_H^{\gamma}$

necessary for the Newton-Raphson linearization can be formed from the following three derivatives:

$$\frac{\partial S_{IJ}^{\alpha\beta^{(n+1)}}}{\partial\psi_H^\gamma}; \qquad \frac{\partial D_{IJ}^{\alpha\beta^{(n+1)}}}{\partial\psi_H^\gamma}; \qquad \frac{\partial F_I^{\alpha^{(n+1)}}}{\partial\psi_H^\gamma}. \tag{0.3}$$

Together with the specifics outlined previously for forming these derivatives, there is enough information to write the pseudocode describing the formation of the Jacobian matrix.

The pseudocode for forming the Newton-Raphson Jacobian matrix is presented here. The code is presented once for the unshifted pass, and once for the shifted pass where the properties have been found for one set of shifted variables. Note the following extra definition:

$$\psi_J^{\beta''} = \frac{\Delta\psi_J^{\beta^n}}{\Delta t_n}. \tag{D.4}$$

Comments are in the style of the C programming language: /* a comment */. All subscripted variables are vectors or matrices. References to actual program vector variables have subscripts with square brackets, e.g. *A[i]*.

D.7.1 *Unshifted pass*

This is the first of two passes to form the Jacobian matrix. This pass will find and disperse all those parts of the derivatives which are not themselves derivatives, and fill the right-hand side vector. The unshifted components of the constituent derivatives are also found.

```
find properties;
find well parametars;
set gsm, rhs, and temprhs vectors to zero;
form $\psi_J^{\beta'}$ and $\psi_J^{\beta''}$ for all $\beta$ variables;
initialize all the level-$n$ storage to zero, for all $\beta$ variables;

loop over $\alpha = 1$,ndftype equations /* ndftype = 3,4 or 7 */
                              /* call the appropriate $\alpha$ equation routine */
    set all the sdgama vectors to zero;
    get the subcoefficients for the $\alpha$ equation;
    form ${F_{IH}^{\alpha}}^{(n+1)}$; /* this also initializes sdfih */
    if still at level-$n$ then /* i.e.  first newton iteration */
        store $\sum_H {F_{IH}^{\alpha}}^n \times (1-\theta)$ in flvln[$\alpha$]; /* n8-long vector */
    store $-flvln[\alpha]$ in temprhs;
    subtract $\theta F_I^{\alpha(n+1)}$ from temprhs;

    loop over $\beta = 1, ndftype$ /* variables, to form $\mathcal{M}_I^\alpha$ */
        loop over $H = 1, nnodes$ /* nnodes = 8 (nodes) */
            form ${S_{IJH}^{\alpha\beta}}^{(n+1)}$; /* n64-long vector */
            add $\theta^2 S_{IJH}^{\alpha\gamma=\beta(n+1)}$ to $sdgama[\gamma = \beta]$; /* n64-long vector */
            if still at level-$n$ then
                add $(1-\theta){S_{IJH}^{\alpha\beta}}^n$ to $slvln[\alpha, \beta]$;
            form ${S_{IJH}^{\alpha\beta}}^{(n+1)} \psi_J^{\beta'}$ and add to $sdfih[H]$;
            subtract $\theta {S_{IJH}^{\alpha\beta}}^{(n+1)} \psi_J^{\beta'}$ from temprhs;

            form ${D_{IJH}^{\alpha\beta}}^{(n+1)}$; /* n64-long vector */
            add $\frac{\theta}{\Delta t_n} D_{IJH}^{\alpha\gamma=\beta(n+1)}$ to $sdgama[\gamma = \beta]$; /* n64-long vector */
            if still at level-$n$ then
                add $(1-\theta){D_{IJH}^{\alpha\beta}}^n$ to $dlvln[\alpha, \beta]$;
            form ${D_{IJH}^{\alpha\beta}}^{(n+1)} \psi_J^{\beta''}$ and add to $sdfih[H]$;
            subtract $\theta {D_{IJH}^{\alpha\beta}}^{(n+1)} \psi_J^{\beta''}$ from temprhs;
        end loop on $H$;

        subtract $(1-\theta){S_{IJ}^{\alpha\beta}}^n \psi_J^{\beta'} + (1-\theta){D_{IJ}^{\alpha\beta}}^n \psi_J^{\beta''}$ from temprhs;
                /* $= \mathcal{M}_I^\alpha$ at end of all $\beta$ */
    end loop on $\beta$;

    loop over $\gamma = 1, ndftype$ /* derivative variables */
        add $sdgama[\alpha, \gamma]$ $+\theta(1-\theta){S_{IH}^{\alpha\gamma}}^n + \frac{(1-\theta)}{\Delta t_n}{D_{IH}^{\alpha\gamma}}^n - \frac{\theta}{\Delta\psi^\gamma} \times sdfih$
                to temporary storage;
                /* $sdgama[\alpha, \gamma] = \theta^2 {S_{IJH}^{\alpha\gamma}}^{(n+1)} + \frac{\theta}{\Delta t_n}{D_{IJH}^{\alpha\gamma}}^{(n+1)}$ */
                /* note that ${F_I^\alpha}^n$ is not in the derivative */
        disperse (add) the temporary storage into gsm
                        according to dof numbering;
        disperse temprhs to rhs according to dof numbering;
    end loop on $\gamma$;
end loop on $\alpha$.
```

D.7.2 *Shifted pass*

The right-hand side vector is now complete. We only need to complete the derivatives in $\mathcal{M}_I^\alpha$.

```
loop over $\gamma = 1, ndftype$ /* shift for each derivative */
    shift $\psi_J^\gamma \rightarrow \psi_J^\gamma + \Delta\psi^\gamma$;
    find properties;
    find well parameters;

    loop over $\alpha = 1, ndftype$ /* equations */
        form $F_{IH}^{\alpha\,(n+1)}$; /* this also initializes sdfih */
        loop over $\beta = 1, ndftype$ /* variables, to form $\mathcal{M}_I^\alpha$ */
            loop over $H = 1, nnodes$ /* nnodes = 8 (nodes) */
                form $S_{IJH}^{\alpha\beta\,(n+1)}$; /* n64-long vector */
                form $S_{IJH}^{\alpha\beta\,(n+1)} \psi_J^{\beta\prime}$ and add to sdfih[H];
                form $D_{IJH}^{\alpha\beta\,(n+1)}$; /* n64-long vector */
                form $D_{IJH}^{\alpha\beta\,(n+1)} \psi_J^{\beta\prime\prime}$ and add to sdfih[H];
            end loop on $H$;
        end loop on $\beta$;

        put $\frac{\theta}{\Delta\psi^\gamma} \times sdfih$ into temporary storage;
        disperse (add) the temporary storage into gsm
                                        according to dof numbering;
    end loop on $\alpha$;
end loop on $\gamma$.
```

D.7.3 *Notes*

This code outline is a trade-off between storage and recomputation and between storage and dispersal. A more compact scheme would be to store all the $n+1$ level variables S and D, but this would double the storage used for the n-level variables S and D. There is considerable overhead associated with accessing the data for large problems with many nodes.

D.8 *Forming the global equations*

The general linearized equation given in Equation (7.4) is:

$$\frac{\partial \mathcal{M}_I^\alpha}{\partial \psi_H^\gamma} \Delta \psi_H^\gamma = -\mathcal{M}_I^\alpha .$$

This equation is only for one finite element in the domain being modelled. As nodes frequently are shared by more than one element, there is coupling of degrees of freedom through compatability—the condition that the value of a degree of freedom, at a node shared by more than one finite element, must be the same for each element. The element equations must be assembled to form a set of global equations dealing with global degree of freedom only. The coefficient matrix for this set of equations is known as the global stiffness matrix (GSM). It is called a stiffness matrix because of the analogy with the matrix methods used for structural analysis before finite elements were well established. The global assembly involves scattering the ESM terms into the GSM and the element right-hand side vector terms into the global right-hand side vector. The dispersal of terms must account for the storage scheme used by the linear solver.

The solution scheme uses the Gaussian solver STAIRWAY. The coefficient matrix is stored in a vector in column order, and stores only data between the bounds of the first and last non-zero elements in the column, plus any elements to satisfy the stairway requirement. The stairway requirement is that there are no re-entrant rows or columns—the profile always resembles a set of (sometimes uneven) stairs. A locator vector is needed to identify the element positions:

- *loc[2*j]* gives the row number of the first non-zero element in column j, which is the first stored element from the column;
- *loc[2*j-1]* gives the starting position of column j in the GSM vector.

Thus we can find any element a_{ij} in *A[sizea]*, where *sizea* is the length of A, from

$$loc\_in\_a = loc[2*j-1] + (i - loc[2*j])$$

$$\Rightarrow a_{ij} = A[loc\_in\_a]$$

D.8.1 *Scattering the* ESM *terms to the* GSM

A generic term in the ESM is

$$\frac{\partial \mathcal{M}_I^\alpha}{\partial \psi_H^\gamma}$$

The Jacobian subroutines return one of these terms for all elements at the completion of each stage of the calculations. It is an `nnodes`×`nnodes` (effectively an 8 × 8) submatrix repeated `nelem` times. The generic term provides the following information:

α indicates the equation type;

I indicates the node number of the degree of freedom;

γ indicates the type of degree of freedom (S_w, S_a, p_w, T, etc.);

H indicates the node number of the multiplying degree of freedom (of type γ).

The position of each element from this ESM term in the GSM will be governed by the following.

a) The equation to which it belongs (by global degree of freedom)—the row number in the GSM. This is the same as the global identification of the degree of freedom type α, the local node number I, and the element number. The degree of freedom type and the local node number are given by the element equation. Each element sharing a node will contribute an equation for each global degree of freedom at that node, and the equations will be summed for the entry into the global equation set.

b) the global degree of freedom by which it is being multiplied—the column number in the GSM. This is the same as the global identification of the degree of freedom type γ, the local node number H, and the element number. The degree of freedom type and the local node number are given by the element equation.

With this information the global degree of freedom number can be found from the element-ordered steering array. The steering array is a table with row entries for the local node and element numbers, and column entries for the degree of freedom type. Scattering data into the right-hand side vector is simpler than for the GSM—only the equation (row) number is needed.

D.8.2 *Pseudocode to scatter to* GSM

One block of data is passed to be scattered. It is stored as in Eqn. (A.5) in blocks of constant H, comprising *nelem* groups of *nnodes* local node data.

```
/* subroutine gsmfill */

given $\alpha$, $\gamma$, and the block of ESM data:
loop over $H = 1$,nnodes /* local columns */
```

```
    loop over $elem = 1, nelem$ /* element numbers */
        find the GSM column from $\gamma$ and $H$;
        if the DOF does not exist ($\equiv 0$), go to the next element;

        loop over $I = 1, nnodes$
            find the GSM row from $\alpha$ and $I$;
            if the DOF does not exist ($\equiv 0$), go to the next $I$;
            find the destination in GSM vector gsm from the locator vector,
                        GSM row, and GSM column;
            increment the destination element by the current element
                        (from $H$, elem, $I$) from the ESM block of data;
        end loop on $I$;
    end loop on elem;
end loop on $H$;
```

D.8.3 *Pseudocode to scatter to* RHS *vector*

One block of data is passed to be scattered. It is stored as *nelem* blocks of *nnodes* local node data.

```
/* subroutine gsmfill */
given $\alpha$ and the block of ESM data:
loop over $elem = 1, nelem$ /* element numbers */
    loop over $I = 1, nnodes$
        find the RHS row rhsrow from $\alpha$ and $I$;
        if the DOF does not exist ($\equiv 0$), go to the next $I$;
        increment rhs[rhsrow] by the current element (elem, $I$) from
                        the ESM block of data;
    end loop on $I$;
end loop on elem;
```

D.9 *Global mesh numbering and the* RENUM *preprocessor*

Due to the mixed (L6 plus L8) element formulation and the ability to approximate irregular boundaries inherent in the finite element method, the best nodal numbering strategy is not immediately obvious. Poor numbering strategies result in global stiffness matrices which have large profiles and bandwidths. These measures are defined by:

$$\left.\begin{aligned} \text{matrix bandwidth} \quad B &= \max_{i \leq N} b_i \\ \text{profile} \quad P &= \sum_{i=1}^{N} b_i \end{aligned}\right\} \tag{D.5}$$

where b_i is the number of columns containing elements of interest (probably non-zero values) in row i, counting from the main diagonal. B is sometimes called the matrix half-bandwidth.

Another contributing factor to the values of B and P is that the nodes may have different degree of freedom (unknowns to be solved for), varying from one to four for the fluid model alone. Systems of linear equations whose coefficient matrices have large bandwidths or profiles require more work for solution. A common reordering scheme for 5- or 7-point finite difference systems is the so-called D4 reordering of Price and Coats (1974). That is a particular scheme for systems with penta- and septadiagonal coefficient matrices. The coefficient matrices arising from the finite element method do not, in general, exhibit such conveniently predictable structures. Instead, a general resequencing program is used which can emphasize minimizing either the profile or the bandwidth.

The core of the program is the ACM TOMS (Association for Computing Machinery Transactions on Mathematical Software) Algorithm 582, produced by Lewis (1982a) for structurally symmetric matrices. This is an extremely efficient implementation of both the Gibbs-Poole-Stockmeyer (Gibbs *et al.*, 1976) and and the Gibbs-King (Gibbs, 1976) algorithms. (Henceforth referred to as the GPSGK, or core, routine.) The GPS algorithm emphasizes bandwidth reduction, and is a particular implementation of the reverse Cuthill-McKee (1969) algorithm. The GK algorithm emphasizes profile reduction. These are all graph-directed minimal degree methods. The assumptions are that the matrix is structurally symmetric and that diagonal pivoting (symmetric row and column interchanges) does not produce numerical instability (Lewis, 1982b). Lewis's contribution has been to improve substantially the data structures used to represent level structures for the graph of the matrix. The development of these algorithms has been coordinated by the US National Aeronautics and Space Administration (NASA) for use with the NASTRAN finite element program (Everstine, 1979).

The preprocessor `RENUM` controls all the operations prior to and after the resequencing calculations, and passes all the necessary information to the resequencing routines. `RENUM` first takes pertinent information about the elements and their nodes, and the variable number of degrees of freedom for each node, and creates a steering array. The steering array interrelates all this information and gives all the degrees of freedom a global numbering. It can then be used to generate a structural template of the non-zero elements in the global stiffness matrix from the individual contributions from the elemental stiffness matrices. This predicted matrix is then reordered into the Harwell storage scheme for sparse matrices, by storing only the non-zero elements by column. Two integer vectors of the column start offsets (a cumulative tally of elements) and the row indices for the non-zero elements are also generated. This can then be passed directly to the GPSGK routine, along with the choice of algorithm (GPS or GK) to be used.

The core routine returns a reordered index and its inverse, as well as information on the new bandwidth and profile. `RENUM` then uses the reordering index to transform the original steering array and recalculate the global stiffness matrix structure as a check. Two integer locator vectors are also generated at this stage for the `DSTAIR` vectorized Gaussian elimination routine. All the necessary information is stored in a file to be read in by FESPS. This information includes the steering array, the reordered index and its inverse, and the `DSTAIR` locator vector (which constitutes a profile of the coefficient matrix).

At this point the user would need only run FESPS with the original numbering scheme and the extra data file from the `RENUM` preprocessor. All reordering takes place inside FESPS and all data is passed back to the user in the original global numbering system via the inverse index. Both reduction methods should be tried, and the better reordering selected.

An arbitrary mesh is shown in Figure D–6, consisting of 91 degrees of freedom over 35 nodes and 16 elements. There are eight L8 elements and eight L6 elements. Note that the L8 elements are deformed into trapezoids in the example. Three degrees of freedom per node are used except for the end faces of the element numbers 1, 5, 9, 10, 4, 8, 12, and 16, which have only two degrees of freedom per node.

The following results were obtained from RENUM:

numbering	*bandwidth*	*profile*
original	49	2231
After profile reduction (GK)	44	2247
After bandwidth reduction (GPS)	39	2221

The improvement in the global stiffness matrix bandwidth is apparent.

The main program in FESPS needs uses the element-ordered steering array generated by `RENUM` to scatter data from the ESM to the GSM. The steering array has column entries corresponding to the type of degree of freedom, and row entries by element number and local node number. The

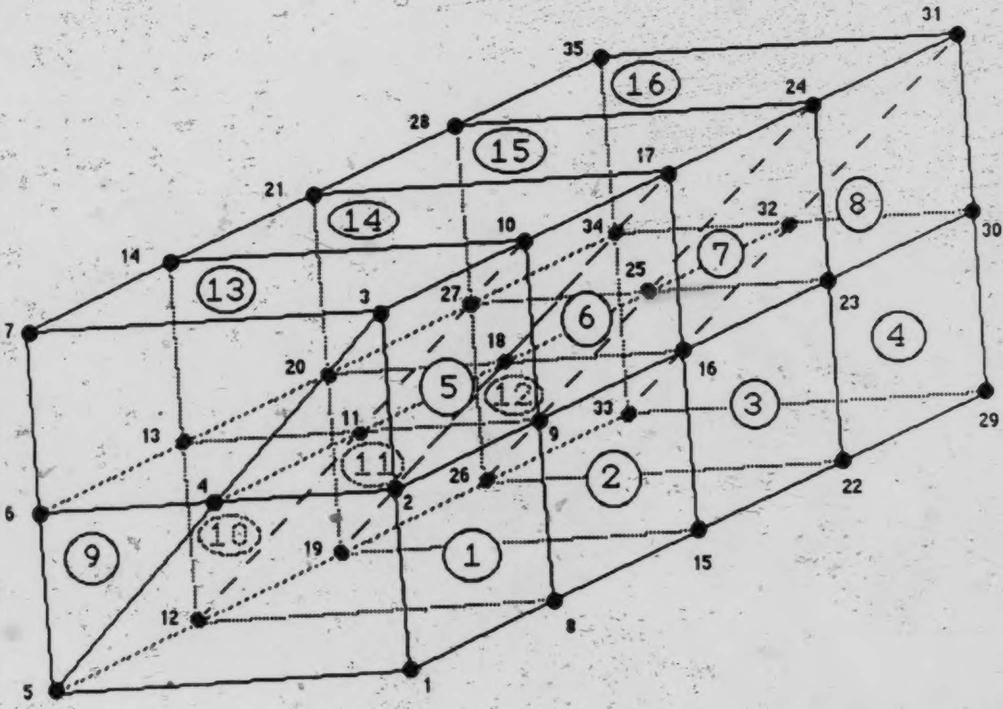

31
35
16
24
28
15
17
21
14
10
34
32
8
30
14
13
3
27
25
7
6
23
4
7
20
18
5
12
16
3
11
13
33
9
29
11
26
2
4
6
22
2
10
9
19
1
15
12
8
5
1

Figure D–6

Test geometry for bandwidth reduction example.

row entries are in groups of elements, with nnodes local nodes for each element. The absence of a particular type of degree of freedom at a local node is denoted by a global degree of freedom value of zero.

D.10 *Vector arithmetic*

This section discusses the mechanics of programming the project. Particular emphasis is laid on the use of the Cyber 205 vector processor because even trivial algorithms must be reworked to take full advantage of the vector pipeline.

The inner product of a matrix and a scalar, $a_{ij}x_j$, where subscripts $i, j = [1..n]$, is usually evaluated as

$$\left\{\begin{matrix} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \\ \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n \end{matrix}\right\}. \tag{D.6}$$

This calculation is inefficient for pipeline processors because it requires $n-1$ summations $n-1$ times. Instead, we redo the calculation so that there are only $n-1$ summations:

$$\left\{\begin{matrix} a_{11} \\ a_{21} \\ \vdots \\ a_{n1} \end{matrix}\right\} x_1 + \left\{\begin{matrix} a_{12} \\ a_{22} \\ \vdots \\ a_{n2} \end{matrix}\right\} x_2 + \cdots + \left\{\begin{matrix} a_{1n} \\ a_{2n} \\ \vdots \\ a_{nn} \end{matrix}\right\} x_n. \tag{D.7}$$

This can also be written as

$$a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n. \tag{D.8}$$

Thus this operation is the addition of n products of a vector and a scalar. There are $n-1$ such additions of the general form $v_1 + v_2 s$, where v_1 and v_2 are vectors, and s is a scalar. This general form is an example of an even more general expression referred to as a linked triad. A linked triad is represented by

$$\textit{(operand}_1\textit{) operator}_1 \textit{ (operand}_2\textit{) operator}_2 \textit{ (operand}_3\textit{)},$$

where two operands must either both be vectors or both be scalars, and the third operand is not the same type as the other pair. The operands must be different and one of the basic arithmetic operations of addition, subraction, multiplication, and division. A linked triad is the most efficient operation on a vector processor that can work in parallel with its unused scalar functional units. Thus this form is sought when designing algorithms for vector arithmetic. In our example, v_1 represents the solution so far for $j = [1..J-1]$, and $v_2 s$ represents $a_{iJ}x_J$.

Generally, the longer the vector, the more effective the vector processor is. The subscripts i and j only vary from 1 to `nnodes` (= 8 usually) so the vector length is quite small. The calculation $a_{ij}x_j$ typically is repeated for each finite element, so we can write our problem to incorporate all the elements in long vectors:

$$a^e_{i1}x^e_1 + a^e_{i2}x^e_2 + \cdots + a^e_{in}x^e_n. \tag{D.9}$$

where e signifies the element number. Thus a^e_{i1} is a_{i1} repeated for every element $e = [1..\texttt{nelem}]$, and $a^e_{i1}x^e_1$ is evaluated as:

$$\begin{array}{c} e = 1 \\ \\ \\ \\ e = 2 \\ \\ \vdots \\ \\ e = nelem \end{array}
\left\{ \begin{array}{c} \left\{\begin{array}{c} a^1_{11} \\ a^1_{21} \\ \vdots \\ a^1_{n1}\end{array}\right\} \\ \left\{\begin{array}{c} a^2_{11} \\ a^2_{21} \\ \vdots \\ a^2_{n1}\end{array}\right\} \\ \vdots \\ \left\{\begin{array}{c} a^e_{11} \\ a^e_{21} \\ \vdots \\ a^e_{n1}\end{array}\right\} \end{array}\right\}
\left\{ \begin{array}{c} \left\{\begin{array}{c} x^1_{1} \\ x^1_{1} \\ \vdots \\ x^1_{1}\end{array}\right\} \\ \left\{\begin{array}{c} x^2_{1} \\ x^2_{1} \\ \vdots \\ x^2_{1}\end{array}\right\} \\ \vdots \\ \left\{\begin{array}{c} x^e_{1} \\ x^e_{1} \\ \vdots \\ x^e_{1}\end{array}\right\} \end{array}\right\}
= \left\{ \begin{array}{c} \left\{\begin{array}{c} a^1_{11}x^1_1 \\ a^1_{21}x^1_1 \\ \vdots \\ a^1_{n1}x^1_1\end{array}\right\} \\ \left\{\begin{array}{c} a^2_{11}x^2_1 \\ a^2_{21}x^2_1 \\ \vdots \\ a^2_{n1}x^2_1\end{array}\right\} \\ \vdots \\ \left\{\begin{array}{c} a^e_{11}x^e_1 \\ a^e_{21}x^e_1 \\ \vdots \\ a^e_{n1}x^e_1\end{array}\right\} \end{array}\right\} . \tag{D.10}$$

Note that only a dyadic product is formed in this example—each pair of numbers is multiplied together. It is not an inner product. Thus the x_j values must be duplicated to fill the subvectors prior to multiplication. This arithmetic leads to the following choice of storage for a_{ij}:

$$\langle a^e_{ij} \rangle = \langle\, \langle a^e_{i1}\rangle, \langle a^e_{i2}\rangle, \langle a^e_{i3}\rangle, \ldots, \langle a^e_{in}\rangle \,\rangle . \tag{D.11}$$

where a^e_{ij} has been laid out as a series of row vectors, each $n \times$ `nelem` long (i varies from 1 to n, for all elements $e = [1..$`nelem`$]$).

Special multiplicative routines are now required to form the various required products. If there is a third index, k, then we string the vectors together for every $k = [1..n]$:

$$\langle a^e_{ijk} \rangle = \langle\, \langle a^e_{ij1}\rangle, \langle a^e_{ij2}\rangle, \ldots, \langle a^e_{ijn}\rangle \,\rangle . \tag{D.12}$$

in a similar fashion to Equation (D.11).

D.10.1 *Multiplication routines*

Certain products arise in the calculations, especially for the Jacobian construction, which are resolved by special routines. These calculations are for:

1. $a_H \Upsilon_{KIJ} \delta_{HK}$, i.e., a dyadic multiplication and no summation;
2. $a_H \Upsilon_{JI} \delta_{HJ}$, i.e., a dyadic multiplication and no summation;

3. $a_J \Upsilon_{JI}$, i.e., an inner product.

These routines are now described.

D. 10. 2 *Multiplying subroutine* KMULT64

This routine evaluates the expression $a_H \Upsilon_{KIJ} \delta_{HK}$ for a given value of K, where the integral is stored as described in §A.1. To form the product, the Kth term for every finite element in the vector a_H must first be extracted. The multipliers a_K^e are formed as in Equation (A.5) by duplicating the a_K^e values to fill the multiplier vector. This vector is then multiplied with the integral Υ_{KIJ} for the particular value of K. `KMULT64` actually cycles through each value of J to form the product.

D. 10. 3 *Multiplying subroutine* JMULT64

This routine completely evaluates the expression $a_H \Upsilon_{JI} \delta_{HJ}$. The Jth term in the vector a_H is extracted and then duplicated to form the multiplying vector in Equation (A.5). The product is now formed for that value of J. This is repeated for every $J = [1..\texttt{nnodes}]$.

D. 10. 4 *Multiplying subroutine* DOTJ64

This routine forms the inner product $a_J \Upsilon_{JI}$. The method is similar to that of the routine `JMULT64`, except that the products formed are added together for all $J = [1..\texttt{nnodes}]$.

D. 11 *Vectorized linear solver* DSTAIR

An effective linear equation solver for general use was needed for FESPS. The expected coefficient matrices would be positive-definite and asymmetric with a symmetric structure and jagged profile. Additionally, an arrangement for mixed elements and degrees of freedom required a general form of storage such as a linear array. The solver would not provide pivoting or scaling of the equations, but would check for zero pivots.

No vectorized linear solvers meeting these criteria existed at the Cyber 205 site when this thesis was first begun. As a result, a general Gaussian solver that would be highly efficient on a vector computer was designed. This solver has the subroutine name DSTAIR, reflecting the profile of the coefficient matrix.

DSTAIR is a column-storage solver, where the columns of the coefficient matrix are stored successively in a vector. The starting position of each column in the vector and its row identity as the first non-zero element in that column are stored in an index vector of length $2N+1$, where N is the number of unknowns. A staircase profile is necessary to avoid computational difficulty during the forward elimination step.

Efficient software on a vector computer must take advantage of the pipeline processors. On the Cyber 205, the fastest arithmetic is achieved by combining one vector operation with one scalar operation over three operands in a sequence called a linked triad, described in §5.7. In the forward elimination stage of Gaussian elimination the linked triad is implemented at each step using outer products. Gentzsch (1984) describes this in detail, but it can be summarized as the implementation of a forward elimination step as a columnwise rather than the usual rowwise operation. This is illustrated in Equations (5.7) and (5.8). During each step information is also stored to operate on the residual vectors during iterative improvement. The working storage used for each step is allocated dynamically during the run.

On completion of forward elimination the upper triangular matrix can be shuffled backwards to release storage in the original coefficient matrix vector. This free storage is used to hold the residual and the incremental solutions for iterative improvement. The backward substitution stage is implemented as a series of linked triads where the solution vector is upgraded at each step by the dyadic addition of a column of numbers.

If iterative improvement or a single residual vector is required the original coefficient matrix must be stored before performing the forward elimination. This can be multiplied in a columnwise manner with the solution vector to obtain the residual with the original right-hand-side vector. The residual is reformed with the stored forward elimination data and then the back substitution stage can proceed as before. The indicated relative error (Rice, 1983) controls the amount of iterative improvement. This is given by the norm of the increment of solution divided by the norm of the previous solution. Intrinsic vector functions in CDC FORTRAN obtain the norms of the vectors.

The comparative performance of DSTAIR as a Gaussian solver has been reported by Tortike and Farouq Ali (1987). Several commercial, general linear-equation solving routines have been made available for research use on the Cyber 205 in Calgary, but do not exhibit the ability to iteratively improve the solution while maintaining high computational speeds. DSTAIR equals the best solvers in all other measures, and is superior to the remaining solvers.

D.12 *Auxiliary equations*

D.12.1 *The material balance*

The flow equations are equations of mass conservation. This fact allows a simple check to be performed on the behaviour of the method of discretization in time. The material balance in its simplest form is that, over the domain Ω with boundaries Γ, component i is conserved:

$$\frac{dm_{i_\Omega}}{dt} + \rho_i q_{i_\Gamma} = 0. \tag{D.13}$$

This equation can be integrated over the timestep used in the time discretization, i.e., time interval $\Delta t_n = t^{n+1} - t^n$,

$$(m_{i_\Omega}^{n+1} - m_{i_\Omega}^{n}) + \int_{t^n}^{t^{n+1}} \rho_i q_{i_\Gamma}\, dt = 0. \tag{D.14}$$

Thus the change in mass of component i over timestep Δt_n should be equal to the time integral of the boundary fluxes over the same time integral. If the mass changes are not equal it indicates the presence of an inconsistancy in the time discretization procedure. The inconsistency might be a function of the size of the time step. The integral $\int_{\Delta t_n} \rho_i q_{i_\Gamma}\, dt$ is evaluated numerically using the same method used for the main set of conservation equations:

$$\int_{\Delta t_n} q_{i_\Gamma}\, dt = \Delta t_n \rho_i \left[(1-\theta) q_{i_\Gamma}^{n} + \theta q_{i_\Gamma}^{n+1}\right]. \tag{D.15}$$

See the section on time integration. The density of the produced fluid ρ_i is constant for produced volumes at a reference condition. The material balance can also be expressed as the ratio

$$\begin{aligned} \frac{\text{net change in mass over time interval } \Delta t_n}{\text{net mass throughput over } \Delta t_n} &= \frac{m_{i_\Omega}^{n+1} - m_{i_\Omega}^{n}}{\Delta t_n \rho_i \left[(1-\theta) q_{i_\Gamma}^{n} + \theta q_{i_\Gamma}^{n+1}\right]} \\ &= -1. \end{aligned} \tag{D.16}$$

In the finite element formulation, the masses in the numerator of Equation (0,16) are weighted approximations. The density is known for the production terms, which are also known exactly for the element. Thus, the equation also includes some error from the weighted approximations in the numerator.

The masses of the components bitumen, gas, and water are found by integrating the pertinent saturations over the problem domain. As there are only piecewise continuous approximations to the variables, the integral over the whole domain is represented by the sum of the integrals over the subdomains, or finite elements.

Bitumen:

$$\int_{\Omega}\left[\phi\frac{\rho_b}{1+R_{so_r}}\right]S_o\,d\Omega=\sum_e\int_{\Omega^e}\left[\phi\frac{\rho_b}{1+R_{so_r}}\right]S_o\,d\Omega. \tag{D.17}$$

Gas:

$$\int_{\Omega}[(\phi\beta_v y_g)S_g+(\phi\rho_g R_{so_r})S_o]\,d\Omega=\sum_e\int_{\Omega^e}[(\phi\beta_v y_g)S_g+(\phi\rho_g R_{so_r})S_o]\,d\Omega. \tag{D.18}$$

Water:

$$\int_{\Omega}[(\phi\alpha_v y_s)S_g+(\phi\rho_w)S_w]\,d\Omega=\sum_e\int_{\Omega^e}[(\phi\alpha_v y_s)S_g+(\phi\rho_w)S_w]\,d\Omega. \tag{D.19}$$

The energy balance has the same form as the mass balance, but note that the net energy throughput must include heat losses to the surrounding formations as well as the well production and injection. The energy content of the domain is made by summing the enthalpy × mass products for each component, including the rock. If the phase densities and enthalpies are known, the enthalpy × mass products can be found for each phase:

$$\begin{aligned}&\int_{\Omega}[(\phi h_o\rho_o)S_o+(\phi h_v\rho_v)S_v+(\phi h_w\rho_w)S_w+((1-\phi)\rho_m)h_m]\,d\Omega\\&\qquad=\sum_e\int_{\Omega^e}[(\phi h_o\rho_o)S_o+(\phi h_v\rho_v)S_v+(\phi h_w\rho_w)S_w+((1-\phi)\rho_m)h_m]\,d\Omega.\end{aligned} \tag{D.20}$$

Note that the solid matrix enthalpy is used as the primary variable for integration. It is a function of temperature only, so it is more appropriate to use than the $(1-\phi)$ term.

The current pore volume can be estimated by assuming that the porosity varies smoothly over each element, then it simply evaluated as

$$\begin{aligned}\int_{\Omega}\phi\,d\Omega&=\sum_e\int_{\Omega^e}\phi\,d\Omega\\&=\phi_I\sum_e\int_{\Omega^e}N_I\,d\Omega\end{aligned} \tag{D.21}$$

The integrals can be evaluated using the basis functions so that, in general, if $S=S(\Omega,t)$ and $A=A(\Omega,t,S)$, then

$$\begin{aligned}\int_{\Omega^e}A(\Omega,t,S)S(\Omega,t)\,d\Omega&=\int_{\Omega^e}\left[\sum_{J=1}^{n}A_J(t,S)N_J(\Omega)\right]\left[\sum_{I=1}^{n}S_I(t)N_I(\Omega)\right]d\Omega\\&=S_I A_J\int_{\Omega^e}N_J N_I\,d\Omega\\&=S_I A_J\Upsilon_{JI}\end{aligned} \tag{D.22}$$

This is the same approach used to derive the spatial approximation to the original flow equations. Thus it can be considered to be consistent with the original problem discretization.

D.13 *Implementation of pseudo-descriptors*

Much of the initial programming of FESPS was done for the CDC Cyber 205 vector processor—in particular, the integration module, the properties module, the linear solver, and the Jacobian module. The rationale behind this approach was that the large problem to be solved required a fast computer with large amounts of virtual memory, so the program should be designed for the computer right from the start. The essential features of this programming include the use of data rearrangement to take advantage of the vector pipelines, and the use of the Cyber FORTRAN200 programming language—particularly the *descriptor* data type. The descriptor data type uses the same amount of storage as a real scalar variable, but refers to a vector of given length. The descriptors are assigned to any variable, and the assignment statement includes the length of the vector being described. Typically, a descriptor is assigned to various portions of vectors or arrays defined during compilation, or it may be assigned storage dynamically during run time. The descriptors are defined with a base type (real, integer, etc.) which must be compatible with the vectors to which they are assigned. The advantage of using a descriptor lies in its compact notation for a vector, and in the fact that it can be used directly in arithmetic statements or passed as an argument to a subprogram. Thus the presence of many do-loops is circumvented. The other vector extensions to ANSI FORTRAN77 (F77) present in Cyber FORTRAN200 include an explicit vector notation, `a(s;n)`, which can be used in arithmetic statements and is analogous to a descriptor describing the same vector start `s` and length `n`. There are vector equivalents of all the standard intrinsic functions, and several other vector functions to facilitate the rearrangement of vector data or to compute things such as the inner product of two vectors. The vector functions take descriptors or explicit vector notation as arguments, and provide the fastest way to evaluate the subprograms described by the functions. The vector functions are conveniently named with the first three letters being either Q8V for functions returning vector values, or Q8S for functions returning scalar values.

The further development of the FESPS program was hindered by the relatively primitive state of software development tools available on the Cyber 205, and the cumbersome access to the machine in Calgary. As a result, further development of the program modules was carried out to conform to ANSI FORTRAN77 so that the more advanced software development tools available on modern Unix workstations could be taken advantage of. While this was successful, the modules already developed could not be integrated with the new code to be compiled under an ANSI FORTRAN77 compiler and run as a complete program. The modules written for the Cyber 205 would have had to have been completely rewritten to be compiled in ANSI FORTRAN77, and, because of the extensive use of descriptors, rewriting the code would be a difficult task as the original form of the programs would be completely lost.

In order to retain the compactness of the Cyber FORTRAN200 descriptor notation, a method was devised of implementing a comparable data type in the ANSI FORTRAN77 environment. All other vector FORTRAN extensions could be emulated by subprograms, or, in the case of explicit vector notation, unravelled using do-loops.

The method is as follows: all variables of FORTRAN200 type descriptor are changed to type double complex (`complex*16`). Vector information from master vectors defined in the main program is stored in the variable as follows:

```
z=(rrr.xxx, iii.yyy)
```

where

`rrr` is the location of the start of the vector, pointed to by `z`, within a master vector;
`iii` is the length of the vector pointed to by `z`;
`xxx` is the master vector number;
yyy is the data type (0=real, 1=integer, 2=bit or logical).

The master vectors are in common blocks as a single integer followed by the master vector, as in the following example:

```
integer kx
real*8 xv(xvlength)
common /xv1/ kx, xv
```

The integer stores the current length of the vector assigned. In this work, only 2 sets of master vectors are used for each type of real, integer, and bit (or logical): no. 1 is for specific assignments; no. 2 is for dynamic allocation which can be freed. Assignment routines are provided for each base type (real, integer, and logical) and for dynamic temporary storage which can be freed. Standard assignments cannot be freed because the book-keeping required would be too great.

The new descriptor implementation, referred to as *pseudo-descriptors*, combined with the emulation routines, permits code written in Cyber FORTRAN200 to be converted quickly and to retain the original structure. The latter advantage is important because there is less likely to be a mistake when converting a given algorithm. The converted code can also be run on the Cyber 205 and because it retains the structure of the vector algorithms used, the original vector performance is retained—at least most of it, there are a couple of overheads in subroutine calls. However, arrays of two or more dimensions have not been supported in this work.

An emulation module comprising the glue routines to make this all work together has been written and tested, although all the Cyber FORTRAN200 Q8Vxxx and Q8Sxxx functions have not, as yet, been incorporated because the emulation routines were only written as they were encountered in the original source code. All arithmetic must be carried out via subroutine calls, as must bit vector testing, and the assignments. Linked triads are provided for by calling a routine with operator information, as are vector **where** statements. The Q8Vxxx and Q8Sxxx calls completed so far are implemented using exactly the same argument order and types as given in the Cyber FORTRAN200 manual. The Q8Vxxx functions are implemented, however, as subroutines. Specific FORTRAN200 code is embedded in all the subroutines and functions, so one can either comment out the scalar code and activate the vector code, or keep two sets of the glue routines: one specifically for FORTRAN77, and one specifically for the Cyber 205.

The conversion is not painless because one must alter every line involving a descriptor or other vector fortran syntax, but the replacements are highly mnemonic, and generally replace just one line. However, the general structure of the source is not at all changed, and the operation should not take very long, particularly with editors supporting regular expressions and macros. The code has so far only been tested in an f77 environment on a Sun 3/160; the Cyber 205 equivalent is a straightforward line in each case.

The implementation of pseudo-descriptors has turned out to be very useful. The type is very compact and makes otherwise long and tortuous source code more understandable as the do-loops are unravelled, and length information need not be passed to subroutines. The pseudo-descriptors can be considered useful in any ANSI FORTRAN77 program where vectors are extensively used and particularly where subroutines are an essential feature.

D. 14 *Pseudo-descriptor subroutines*

The following subroutines perform arithmetic operations on pseudo-descriptors. The subscript `r` refers to the version requiring real arguments, `i` refers to integer arguments, and `b` to logical (bit) arguments. It is understood here that the explanations given refer to the values of the vectors described by the descriptors. This is also indicated by the notation `descriptor->vector`. All the operations assume double precision variables (`real*8`).

a) Descriptor-descriptor arithmetic operations (`result`, `a`, and `b` are descriptors):

`multr (product, a, b):  product->vec = a->vec * b->vec;`

`multi (product, a, b):  product->vec = a->vec * b->vec;`

`divr (quotient, a, b):  quotient->vec = a->vec / b->vec;`

`divi (quotient, a, b):  quotient->vec = a->vec / b->vec;`

`addr (sum, a, b):  sum->vec = a->vec + b->vec;`

`addi (sum, a, b):  sum->vec = a->vec + b->vec;`

`subr (diff, a, b):  diff->vec = a->vec - b->vec;`

`subi (diff, a, b):  diff->vec = a->vec - b->vec;`

b) Descriptor-scalar arithmetic operations (`result`, `a` are descriptors; `b` is a scalar):

`smultr (product, a, b):  product->vec = a->vec * b;`

`smulti (product, a, b):  product->vec = a->vec * b;`

`sdivr (quotient, b, a):  quotient->vec = a->vec / b;`

`sdivi (quotient, a, b):  quotient->vec = a->vec / b;`

`saddr (sum, a, b):  sum->vec = a->vec + b;`

`saddi (sum, a, b):  sum->vec = a->vec + b;`

`ssubr (diff, a, b):  diff->vec = a->vec - b;`

`subrs (diff, b, a):  diff->vec = b - a->vec;`

`ssubi (diff, a, b):  diff->vec = a->vec - b;`

`subis (diff, b, a):  diff->vec = b - a->vec;`

c) Descriptor-descriptor manipulations (`object` and `x` are descriptors; `offset` is an integer):

`vvsetr (object, x):  object->vec = x->vec;`

`vvseti (object, x):  object->vec = x->vec;`

`vvsetb (object, x):  object->vec = x->vec;`

`vvofsetr (object, x, offset):  object->vec(+offset) = x;`

`vvofseti (object, x, offset):  object->vec(+offset) = x;`

`vvofsetb (object, x, offset):  object->vec(+offset) = x;`

d) Descriptor-scalar arithmetic operations (`object` is a descriptor; `x` is a scalar):

```
vssetr (object, x):  object->vec = x;

vsseti (object, x):  object->vec = x;

vssetb (object, x):  object->vec = x;
```

e) Descriptor-vector manipulations up to length of descriptor operand (`object` is a descriptor; `x` is a vector):

```
pvsetr (object, x):  object->vec = x;

pvseti (object, x):  object->vec = x;

pvsetb (object, x):  object->vec = x;
```

f) Vector-descriptor manipulations up to length of descriptor operand (`object` is a vector; `x` is a descriptor):

```
vpsetr (object, x):  object = x->vec;

vpseti (object, x):  object = x->vec;

vpsetb (object, x):  object = x->vec;
```

g) Manipulations of elements of a descriptor (`object` is a descriptor; `loc` is an integer, `value` is a scalar):

```
setvalr (object, loc, value):  object->vec(loc) = value;

setvali (object, loc, value):  object->vec(loc) = value;

real*8 function getvalr (object, loc):  getvalr = object->vec(loc);

integer function getvali (object, loc):  getvali = object->vec(loc);
```

D.15 *Cyber* FORTRAN*200 functions and their emulations*

a) Scalar functions corresponding to intrinsic vector functions and Q8Sxxx functions in Cyber FORTRAN200:

```
integer function q1slen (v):  q8slen(v);

integer function q1scnt (v):  q8scnt(v);

real*8 function q1smaxr (v):  q8smax(v);
```

`integer function q1smaxi (v):  q8smax(v);`

`integer function qsmaxloc (v):  q8smaxi(v);`

`real*8 function q1sdot (u, v):  q8sdot(u;v);`

b) Descriptor functions corresponding to intrinsic vector operations and Q8Vxxx functions in Cyber FORTRAN200:

`q1vcmprs (v, cv, u):  q8vcmprs(v,cv;u);`

`q1vctrl (v, cv, u):  q8vctrl(v,cv;u);`

`q1vgathp (v, period, times, u):  q8vgathp(v,period,times;u);`

`q1vgathr (v, index, u):  q8vgathr(v,index;u);`

`q1vmerg (v1, v2, cv, u):  q8vmerg(v1,v2,cv;u);`

`q1vmko (a1, a2, u):  q8vmko (a1,a2;u);`

`q1vmkz (a1, a2, u):  q8vmkz (a1,a2;u);`

`q1vscatp (v, period, times, u):  q8vscatp(v,period,times;u);`

`q1vscatr (v, index, u):  q8vscatr(v,index;u);`

`q1vxpnd (v, cv, u):  q8vxpnd(v,cv;u);`

`q1vexp (v, u):  vexp(v;u);`

`q1vlog (v, u):  vlog(v;u);`

`q1vabs (v, u):  vabs(v;u);`

`powr (u, v, exponent)`: `u = v**exponent` (exponent scalar);

`spowr (u, v, exponent)`: `u = v**exponent` (v scalar);

c) Descriptor i/o utilities:

`vwrite (output, format, x)`: write the contents of `x->vec` to logical output unit output according to the character string `format`;

`vread (input, format, x)`: read the contents of `x->vec` from logical input unit `input` according to the character string `format`;

`bwrite (output, *, x)`: write the contents of `x->vec` to logical output unit output in binary format; a write error causes the execution to continue at statement label `*`;

`bread (input, *, *, x, reqlen)`: read the contents of `x->vec` from logical input unit `input` in binary format; a read error causes the execution to continue at the first statement label `*`, an end-of-file error at the second statement label `*`;

d) Linked triad emulations to obtain benefits of Cyber 205. The emulation routines emulate the following expressions, noting that the subscript `r` is for real arguments, and subscript `i` is for integer arguments, and that `a` and `b` are scalars for `ssv` and that only `a` is a scalar for `svv`

```
brace = 1:  result = (a [op] b) [op] c

brace = 2:  result = a [op] (b [op] c)

ssvr (result, a, op1, b, op2, c, brace);

ssvi (result, a, op1, b, op2, c, brace);

svvr (result, a, op1, b, op2, c, brace);

svvi (result, a, op1, b, op2, c, brace);
```

e) Assigning pseudo-descriptors.

`rassign (desc, length, vec)`: assign a pseudo-descriptor to a real vector of length `length`, master vector no. `vec`;

`iassign (desc, length, vec)`: assign a pseudo-descriptor to an integer vector of length `length`, master vector no. `vec`;

`bassign (desc, length, vec)`: assign a pseudo-descriptor to a logical (bit) vector of length `length`, master vector no. `vec`;

`assover (desc, length, samedesc, offset)`: assign a pseudo-descriptor to a portion of another pseudo-descriptor's vector, offset by `offset` and of length `length`;

`dynamic (desc, length, vectype)`: assign dynamic storage of type `vectype` (0=real, 1=integer, 2=logical) and length `length`;

`free`: free all dynamic storage.

f) Creating logical vectors by comparing one pseudo-descriptor with another. The `test` is a valid FORTRAN logical comparison mnemonic: `eq`, `ne`, `lt`, `le`, `gt`, or `ge`. In the `vstest` version, y is a scalar, otherwise all the arguments other than the `test` is a pseudo-descriptor. The subscripts r, i, and b are respectively the real, integer and logical versions.

```
vstestr (result, x, test, y);
```

```
vstesti (result, x, test, y);

vstestb (result, x, test, y);

vvtestr (result, x, test, y);

vvtesti (result, x, test, y);

vvtestb (result, x, test, y);
```

g) The following routines emulate the vector statement `where (cv) u = v` in Cyber FORTRAN-200 as follows. The first routine applies to all pseudo-descriptors, while the last three compare a pseudo-descriptor, `u`, with a scalar `v`, the appropriate suffix `r`, `i`, and `b` for real, integer, and logical respectively.

```
wherev (cv, u, v);

wheresr (cv, u, vr);

wheresi (cv, u, vi);

wheresb (cv, u, vb);
```

Appendix E

A HIGHLY-IMPLICIT FINITE DIFFERENCE STEAM MODEL (DEAD OIL) SIM3D3P

This appendix outlines the formulation of a finite difference steam simulator based on the model described by Ferrer *et al.* (1980). The original simulator program was developed by S.M. Farouq Ali, who has tested and used it over the years. The program was then modified to accommodate the coupling to the finite element model, and was tested as a stand-alone simulator to ensure that it still functioned like the original code.

The following difference equations are used for the steam injection model, where steam is the only gas present, but first note the following abbreviations:

$$\delta f = f^{n+1} - f^{n} \tag{E.1a}$$

$$\Delta() = \Delta_x() + \Delta_y() + \Delta_z() \tag{E.1b}$$

Molar balance for oil

$$\Delta[T_o(\Delta p_o - \gamma_o \Delta z)] - q_o^* \rho_o^* = \frac{V_b}{\Delta t}\delta(\phi \rho_o S_o) \tag{E.2}$$

Molar balance for water

$$\Delta[T_w(\Delta p_w - \gamma_w \Delta z) + T_g(\Delta p_g - \gamma_g \Delta z)] - q_w^* \rho_w^* - q_g^* \rho_g^* = \frac{V_b}{\Delta t}\delta\left[\phi(\rho_w S_w + \rho_g S_g)\right] \tag{E.3}$$

Energy balance

$$\Delta[T_H \Delta p] + \Delta[T_c \Delta T] - q_L^* - q_H^* = \frac{V_b}{\Delta t}\delta\left[\phi(\rho_w S_w U_w + \rho_g S_g U_g + \rho_o S_o U_o) + (1-\phi)(\rho C_p)_r T\right] \tag{E.4}$$

Vapour-Liquid (steam-condensate) Equilibrium

$$T = f(p_g) \tag{E.5}$$

We will expand the accumulation terms and iterate on the respective variables (p, T). The transmissibility terms (at $n+1$) are expanded in time, as in the semi-implicit method.

E. 1 *Expansion of the equations (accumulation terms)*

It is assumed that ρ is a function of p and T. The subscript $_p$ indicates a derivative with respect to p, and $f_p^{\ell} = f_p^{n+1}$, the latest iterate. For example

$$k_{ros_w} \equiv \frac{\partial k_{ro}}{\partial S_w} \tag{E.6}$$

Furthermore, the following expansion is defined:

$$\delta f = f_T^{\ell}\delta T + f_p^{\ell}\delta p + f^{\ell} - f^{n} - f_T^{\ell}(T^{\ell} - T^{n}) - f_p^{\ell}(p^{\ell} - p^{n}) \tag{E.7}$$

Thus one obtains,

$$\delta p = \rho_p \delta p + \rho_T \delta T \tag{E.8}$$

$$f^{n+1} = f^{\ell} + f_T^{\ell}(T^{n+1} - T^{n}) + f_p^{\ell}(p^{n+1} - p^{n}) \tag{E.9}$$

$$\begin{aligned} \delta(\phi\rho S) &= \phi^{n+1}S^{n}[\rho^{\ell} - \rho^{n} - \rho_T(T^{\ell} - T^{n}) - \rho_p(p^{\ell} - p^{n})] + \phi^{n+1}\rho^{n+1}\delta S \\ &\quad + (\phi^{n+1}S^{n}\rho_T\delta T + \phi^{n+1}S^{n}\rho_p + \rho^{n}S^{n}\phi_i c_r)\delta p \end{aligned} \tag{E.10}$$

Also,

$$\begin{aligned} \delta(\phi\rho S U) &= U^{n+1}\delta(\phi\rho S) + (\phi\rho S)^{n}\frac{\partial U}{\partial T}\delta T \\ &= U^{n+1}\phi^{n+1}S^{n}\left[\rho^{\ell} - \rho^{n} - \rho_T(T^{\ell} - T^{n}) - \rho_p(p^{\ell} - p^{n})\right] \\ &\quad + \left[\phi^{n+1}U^{n+1}S^{n}\rho_T + (\phi\rho S)^{n}\frac{\partial U}{\partial T}\right]\delta T + \phi^{n+1}\rho^{n+1}U^{n+1}\delta S \\ &\quad + U^{n+1}(\phi^{n+1}S^{n}\rho_p + \rho^{n}S^{n}\phi_i c_r)\delta p \end{aligned} \tag{E.11}$$

Note the following, for $f = f(p, T)$. (We are expanding f^{n+1} with respect to the iteration level.)

$$f^{n+1} = f_{n+1}^{\ell+1} = f_{n+1}^{\ell} + \left.\frac{\partial f}{\partial T}\right|^{\ell}\delta T^{\ell} + \left.\frac{\partial f}{\partial p}\right|^{\ell}\delta p^{\ell} \tag{E.12}$$

Note that

$$\left.\frac{\partial f}{\partial T}\right|^{\ell} = \frac{\partial f}{\partial T}\text{ evaluated at }T^{n+1}$$

Thus,

$$\begin{aligned} f^{n+1} &= f_{n+1}^{\ell} + \left.\frac{\partial f}{\partial T}\right|^{\ell}\left(T_{n+1}^{\ell+1} - T_{n+1}^{\ell}\right) + \left.\frac{\partial f}{\partial p}\right|^{\ell}\left(p_{n+1}^{\ell+1} - p_{n+1}^{\ell}\right) \\ &= f^{\ell} + f_T^{\ell}(T^{n+1} - T^{\ell}) + f_p^{\ell}(p^{n+1} - p^{\ell}) \end{aligned} \tag{E.13}$$

Therefore,

$$\begin{aligned}\delta f &= f^{n+1} - f^{n} \\ &= f^{\ell} + f_T^{\ell}(T^{n+1} - T^{\ell}) + f_p^{\ell}(p^{n+1} - p^{\ell}) - f^{n} \\ &\quad - f_T^{\ell}T^{n} + f_T^{\ell}T^{n} - f_p^{\ell}p^{n} + f_p^{\ell}p^{n} \\ &= f_T^{\ell}\delta T + f_p^{\ell}\delta p + f^{\ell} - f^{n} - f_T^{\ell}(T^{\ell} - T^{n}) - f_p^{\ell}(p^{\ell} - p^{n})\end{aligned} \tag{E.14}$$

where $\delta T = T^{n+1} - T^{n}$ and $\delta p = p^{n+1} - p^{n}$. Here, T^{n+1} and p^{n+1} are really the new iterates of T^{n+1} and p^{n+1}, viz. $T^{\ell+1}_{n+1}$, $p^{\ell+1}_{n+1}$.

E. 2 *Energy balance:*

Consider the energy balance equation first:

$$\begin{aligned}&\delta[\phi(\rho_o S_o U_o + \rho_w S_w U_w + \rho_g S_g U_g) + (1-\phi)(\rho C_p)_r T] \\ &\quad = \phi^{n+1} U_g^{n+1} S_g^{n} [\rho_g^{\ell} - \rho_g^{n} - \rho_{g_T}(T^{\ell} - T^{n}) - \rho_{g_p}(p^{\ell} - p^{n})] \\ &\qquad + \left[\phi^{n+1} \sum_{m=o,w,g} U_m^{n+1} S_m^{n} \rho_{m_T} + \phi^{n} \sum_{m=o,w,g} \rho_m^{n} S_m^{n} \frac{\partial U_m}{\partial T}(1 - \phi^{n+1})(\rho C_p)_r\right] \delta T \\ &\qquad + \phi^{n+1}[\rho_g^{n+1} U_g^{n+1} - \rho_o^{n+1} U_o^{n+1}]\delta S_g + \phi^{n+1}[\rho_w^{n+1} U_w^{n+1} - \rho_o^{n+1} U_o^{n+1}]\delta S_w \\ &\qquad + \left\{\phi^{n+1} \sum_{m=o,w,g} U_m^{n+1} S_m^{n} \rho_{m_p} + \phi_i c_r \left[\sum_{m=o,w,g} \rho_m^{n} S_m^{n} U_m^{n+1} - (\rho C_p)_r T^{n}\right]\right\} \delta p\end{aligned} \tag{E.15}$$

Convection terms:

$$T_H \Delta p = \sum_{m=o,w,g} T_m H_m (\Delta p_m - \gamma_m \Delta z) \tag{E.16}$$

Consider the term $T_o H_o(\Delta p_o - \gamma_o \Delta z)$:

Case (i) No gas in the upstream block (w.r.t. a given phase); implicit in p, T, and S.

$$\begin{aligned}[T_o H_o(\Delta p_o - \gamma_o \Delta z)]^{n+1} &= [T_o H_o(\Delta p_o - \gamma_o \Delta z)]^{n} + T_o^{n} H_o^{n} \Delta\delta p \\ &\quad + (\Delta p_o^{n+1} - \gamma_o \Delta z)\left(T_o^{n} H_{o_T}\delta T + H_o^{n} T \frac{\rho_o}{\mu_o} k_{ro_{S_w}} \delta S_w\right) \\ &= [T_o H_o(\Delta p_o - \gamma_o \Delta z)]^{n} + T_o^{n} H_o^{n} \Delta\delta p \\ &\quad + (\Delta p_o^{n+1} - \gamma_o \Delta z) T_o^{n} H_{o_T} \delta T \\ &\quad + (\Delta p_o^{n+1} - \gamma_o \Delta z) H_o^{n} T \frac{\rho_o}{\mu_o} k_{ro_{S_w}} \delta S_w\end{aligned} \tag{E.17}$$

Notice that the time expansion is in saturation and temperature. $T\frac{\rho_o}{\mu_o}$ is the constant part of the transmissibility which is at n.

Consider the term $T_w H_w(\Delta p_w - \gamma_w \Delta z)$:

Case (i) No gas in the upstream block w.r.t. the water phase.

$$\begin{aligned}
&[T_w H_w(\Delta p_w - \gamma_w \Delta z)]^{n+1} \\
&\quad = [T_w H_w(\Delta p_w - \gamma_w \Delta z)]^n + T_w^n H_w^n \Delta\delta p + (\Delta p_w^{n+1} - \gamma_w \Delta z) T_w^n H_{w_T} \delta T \\
&\qquad - T_w^n H_w^n \Delta P'_{c_{ow}} \delta S_w + (\Delta p_w^{n+1} - \gamma_w \Delta z) H_w^n T \frac{\rho_w}{\mu_w} k'_{rw} \delta S_w \\
&\quad = [T_w H_w(\Delta p_w - \gamma_w \Delta z)]^n + (\Delta p_w^{n+1} - \gamma_w \Delta z) T_w^n H_{w_T} \delta T \\
&\qquad + (\Delta p_w^{n+1} - \gamma_w \Delta z) H_w^n T \frac{\rho_w}{\mu_w} k'_{ro_w} \delta S_w - T_w^n H_w^n \Delta P'_{c_{ow}} \delta S_w + T_w^n H_w^n \Delta\delta p
\end{aligned} \tag{E.18}$$

$$\begin{aligned}
T_H \Delta p = {} & \sum_{m=o,w} [T_m H_m(\Delta p_m - \gamma_m \Delta z)]^n + \left[\sum_{m=o,w} (\Delta p_m - \gamma_m \Delta z)^n T_m^n H_{m_T} \right] \delta T \\
& + \left[T \sum_{m=o,w} (\Delta p_m^{n+1} - \gamma_m \Delta z) H_m^n \frac{\rho_m}{\mu_m} k_{rms_w} \right] \delta S_w \\
& - T_w^n H_w^n \Delta P'_{c_{ow}} \delta S_w + \left(\sum_{m=o,w} T_m^n H_m^n \right) \Delta\delta p
\end{aligned} \tag{E.19}$$

Case (ii) Gas is present in the upstream (w.r.t. a particular phase) block. The convective term for a particular phase is obtained implicitly in S_g, S_w, and p.

Consider the term $T_o H_o(\Delta p_o - \gamma_o \Delta z)$:

$$\begin{aligned}
[T_o H_o(\Delta p_o - \gamma_o \Delta z)]^{n+1} &= [T_o H_o(\Delta p_o - \gamma_o \Delta z)]^n + H_o^n \delta [T_o(\Delta p_o - \gamma_o \Delta z)] \\
&= [T_o H_o(\Delta p_o - \gamma_o \Delta z)]^n \\
&\quad + H_o^n (\Delta p_o^{n+1} - \gamma_o \Delta z) T \left(\frac{\rho_o}{\mu_o} \right)^n k'_{ros_g} \delta S_g \\
&\quad + H_o^n (\Delta p_o^{n+1} - \gamma_o \Delta z) T \left(\frac{\rho_o}{\mu_o} \right)^n k'_{ros_w} \delta S_w + H_o^n T_o^n \Delta\delta p
\end{aligned} \tag{E.20}$$

Consider the term $T_w H_w(\Delta p_w - \gamma_w \Delta z)$:

$$\begin{aligned}
[T_w H_w(\Delta p_w - \gamma_w \Delta z)]^{n+1} &= [T_w H_w(\Delta p_w - \gamma_w \Delta z)]^n + H_w^n \delta [T_w(\Delta p_w - \gamma_w \Delta z)] \\
&= [T_w H_w(\Delta p_w - \gamma_w \Delta z)]^n + H_w^n (\Delta p_w^{n+1} - \gamma_w \Delta z) T \left(\frac{\rho_w}{\mu_w} \right)^n k'_{r_w} \delta S_w \\
&\quad - H_w^n T_w^n \Delta (P'_{c_{ow}} \delta S_w) + H_w^n T_w^n \Delta\delta p
\end{aligned} \tag{E.21}$$

Consider the term $T_g H_g(\Delta p_g - \gamma_g \Delta z)$:

$$[T_g H_g(\Delta p_g - \gamma_g \Delta z)]^{n+1} = [T_g H_g(\Delta p_g - \gamma_g \Delta z)]^n + H_g^n \delta [T_g(\Delta p_g - \gamma_g \Delta z)]$$

$$
\begin{aligned}
&= [T_g H_g(\Delta p_g - \gamma_g \Delta z)]^n + H_g^n(\Delta p_g^{n+1} - \gamma_g \Delta z) T \left(\frac{\rho_g}{\mu_g}\right)^n k'_{r_g} \delta S_g \\
&\quad + H_g^n T_g^n \Delta(P'_{c_{go}} \delta S_g) + H_g^n T_g^n \Delta \delta p
\end{aligned} \tag{E.22}
$$

Thus,

$$
\begin{aligned}
T_H \Delta p = {} & \sum_{m=o,w,g} [T_m H_m(\Delta p_m^{n+1} - \gamma_m \Delta z)]^n + \left[\sum_{m=o,w,g} (\Delta p_m^{n+1} - \gamma_m \Delta z) T_m^n H_{m_T} i_{T_m}\right] \delta T \\
& + \left[\sum_{m=o,g} H_m^n (\Delta p_m^{n+1} - \gamma_n \Delta z) T \left(\frac{\rho_m}{\mu_m}\right)^n k_{rms_g} i_{G_m}\right] \delta S_g + H_g^n T_g^n \Delta(P_{c_{go}} \delta S_g) \\
& + \delta S_w \sum_{m=o,w} H_m^n (\Delta p_m^{n+1} - \gamma_m \Delta z) T \left(\frac{\rho_m}{\mu_m}\right)^n k_{rms_w} i_T + \sum_{m=o,w,g} H_m^n T_m^n \Delta \delta p
\end{aligned} \tag{E.23}
$$

$$
T_c \Delta T = T_c(T_{i-1}^n + \alpha_{i-1} \delta T_{i-1} - T_{i,n} - \alpha_i \delta T_i) \tag{E.24}
$$

$$
q_H = \sum_{m=o,w,g} q_m \rho_m H_m \tag{E.25}
$$

If free gas is present, the enthalpies are treated explicitly in T.

$$
q_H^{n+1} = q_H^n + \sum_{m=o,w,g} \rho_m^n H_m^n \delta q_m \tag{E.26}
$$

$$
q_m^{n+1} = q_m^n + \mathrm{PI}\left[\lambda_m^n \delta p + (p_m^{n+1} - p_{w_b}) \lambda_{ms_m} \delta S_m\right]; \quad m = w, g \tag{E.27}
$$

$$
q_o^{n+1} = q_o^n + \mathrm{PI}\left[\lambda_o^n \delta p + (p_o^{n+1} - p_{w_b})(\lambda_{os_w} \delta S_w + \lambda_{os_g} \delta S_g)\right] \tag{E.28}
$$

$$
q_g^{n+1} = q_g^n + \mathrm{PI}\left[\lambda_g^n \delta p + (p_g^{n+1} - p_{w_b}) \lambda_{gs_g} \delta S_g\right] \tag{E.29}
$$

$$
\begin{aligned}
q_H^{n+1} = {} & \sum_{m=o,w,g} \rho_m^n H_m^n q_m^n + \mathrm{PI}\left(\sum_{m=o,w,g} \lambda_{m_n} \rho_{m_n} H_{m_n}\right) \delta p \\
& + \mathrm{PI}\left[\sum_{m=o,g} \rho_m^n H_m^n (p_m^{n+1} - p_{w_b}) \lambda_{ms_g} \delta S_g + \sum_{m=o,w} \rho_m^n H_m^n (p_m^{n+1} - p_{w_b}) \lambda_{ms_w} \delta S_w\right]
\end{aligned} \tag{E.30}
$$

If free gas is not present, $q_g = 0$, and the enthalpies are treated implicitly in T:

$$
q_H^{n+1} = q_H^n + \sum_{m=o,w} \rho_m^n H_m^{n+1} \delta q_m + \sum_{m=o,w} \rho_m^n q_m^n H_{m_T} \delta T \tag{E.31}
$$

$$
\begin{aligned}
&= q_H^n + \left(\sum_{m=o,w} \rho_m^n q_m^n H_{m_T}\right) \delta T + \mathrm{PI} \sum_{m=o,w} \rho_m^n H_m^{n+1} \lambda_m^n \delta p \\
&\quad + \mathrm{PI}\left(\sum_{m=o,w} \rho_m^n H_m^{n+1} \lambda_{ms_w}\right) \delta S_w
\end{aligned} \tag{E.32}
$$

If water is absent, $S_w = 0$.

$$
\begin{aligned}
q_H^{n+1} &= q_H^n + \sum_{m=o,g} \rho_m^n q_m^n H_{m_T} \delta T + \mathrm{PI} \sum_{m=o,g} \rho_m^n H_m^{n+1} \lambda_m^n \delta p \\
&\quad + \mathrm{PI} \left(\sum_{m=o,g} \rho_m^n H_m^{n+1} \lambda_{m S_g} \right) \delta S_g
\end{aligned}
\tag{E.33}
$$

Finally, the energy balance can be written as follows:

$$
\begin{aligned}
&\Delta \left\{ \sum_{m=o,w,g} [T_m H_m (\Delta p_m - \gamma_m \Delta z)] \right\} \delta T + \Delta \left[\sum_{m=o,w,g} (\Delta p_m^{n+1} - \gamma_m \Delta z) T_m^n H_{m_T} i_{T_m} \right] \delta T \\
&\quad + \Delta \left[\sum_{m=o,g} H_m^n (\Delta p_m^{n+1} - \gamma_m \Delta z) T \left(\frac{\rho_m}{\mu_m} \right)^n k_{rm S_g} i_{G_m} \right] \delta S_g + \Delta \left[H_g^n T_g^n \Delta (P'_{c_{go}} \delta S_g) \right] \\
&\quad + \Delta \left[\sum_{m=o,w} H_m^n (\Delta p_m^{n+1} - \gamma_m \Delta z) T \left(\frac{\rho_m}{\mu_m} \right)^n k_{rm S_w} i_{w_m} \right] \delta S_w \\
&\quad + \Delta \left[\sum_{m=o,w,g} H_m^n T_m^n \Delta \delta p \right] + \Delta \{ T_c \Delta (T_n + i_T \delta T) - \Delta [H_w^n T_w^n \Delta (P'_{c_{ow}} \delta S_w)] \} \\
&\quad - \sum_{m=o,w,g} \rho_m^n H_m^n q_m^n - \left(\sum_{m=o,w} \rho_m^n q_m^n H_{m_T} \right) i_T \delta T - \mathrm{PI} \left[\sum_{m=o,g} \rho_m^n H_m^n (p_m^{n+1} - p_{w_b}) \lambda_{m S_g} i_G i W \right] \delta S_g \\
&\quad - \mathrm{PI} \left[\sum_{m=o,g} \rho_m^n H_m^{n+1} (p_m^{n+1} - p_{w_b}) \lambda_{m S_g} i_G i_T \right] \delta S_g \\
&\quad - \mathrm{PI} \left[\sum_{m=o,w} \rho_m^n H_m^n (p_m^{n+1} - p_{w_b}) \lambda_{m S_w} i W i_G \right] \delta S_w \\
&\quad - \mathrm{PI} \left[\sum_{m=o,w} \rho_m^n H_m^{n+1} (p_m^{n+1} - p_{w_b}) \lambda_{m S_w} i W i_T \right] \delta S_w \\
&\quad - \mathrm{PI} \left(\sum_{m=o,w,g} \lambda_m^n \rho_m^n H_m^n \right) i_G i W \delta p - \mathrm{PI} \sum_{m=o,w,g} \lambda_m^n \rho_m^n H_m^{n+1} i_m \delta p - q_L^n - \alpha \delta T \\
&= \frac{V_b}{\Delta t} \Big\{ \phi^{n+1} U_g^{n+1} S_g^{n+1} \left[\rho_g^\ell - \rho_g^n - \rho_{g_T} (T^\ell - T^n) - \rho_{g_p} (p_g^\ell - p_g^n) \right] \\
&\quad + [\phi^{n+1} \sum_{m=o,w,g} U_m^{n+1} S_m^n \rho_{m_T} + \phi^n \sum_{m=o,w,g} \rho_m^n S_m^n \frac{\partial U_m}{\partial T} + (1 - \phi^{n+1}) (\rho C_p)_r] \delta T \\
&\quad + \phi^{n+1} \left[\rho_g^{n+1} U_g^{n+1} - \rho_o^{n+1} U_o^{n+1} \right] \delta S_g + \phi^{n+1} \left[\rho_w^{n+1} U_w^{n+1} - \rho_o^{n+1} U_o^{n+1} \right] \delta S_w \\
&\quad + \left(\phi^{n+1} \sum_{m=o,w,g} U_m^{n+1} S_m^n \rho_{m_p} + \phi_i c_r \left[\sum_{m=o,w,g} \rho_m^n S_m^n U_m^{n+1} - (\rho C_p)_r T^n \right] \right) \delta p \Big\}
\end{aligned}
\tag{E.34}
$$

The above equation can be written as

$$\sum_{j=1}^{4} C_{1j} p_j = R_1 + \sum_{j=1}^{4} \Delta(T_{1j} \Delta p_j) \tag{E.35}$$

where

$$p_1 = \delta T \tag{E.36a}$$

$$p_2 = \delta S_g \tag{E.36b}$$

$$p_3 = \delta S_w \tag{E.36c}$$

$$p_4 = \delta p \tag{E.36d}$$

and,

$$\begin{aligned}
C_{11} = {} & \frac{V_b}{\Delta t}\left[\phi^{n+1} \sum_{m=o,w,g} U_m^{n+1} S_m^n \rho_{m_T} + \phi^n \sum_{m=o,w,g} \rho_m^n S_m^n \frac{\partial U_m}{\partial T} + (1 - \phi^{n+1})(\rho C_p)_r\right] \\
& + \left(\sum_{m=o,w} \rho_m^n q_m^n H_{m_T}\right) i_T + \alpha
\end{aligned} \tag{E.37a}$$

$$\begin{aligned}
C_{12} = {} & \frac{V_b}{\Delta t}\phi^{n+1}(\rho_g^{n+1} U_g^{n+1} - \rho_o^{n+1} U_o^{n+1}) \\
& + \mathrm{PI}\left[\sum_{m=o,g} \rho_m^n \lambda_{m_{S_g}} (p_m^{n+1} - p_{w_b})(i_G i_W H_m^n + i_G i_T H_m^{n+1})\right]
\end{aligned} \tag{E.37b}$$

$$\begin{aligned}
C_{13} = {} & \frac{V_b}{\Delta t}\phi^{n+1}(\rho_w^{n+1} U_w^{n+1} - \rho_o^{n+1} U_o^{n+1}) \\
& + \mathrm{PI}\left[\sum_{m=o,w} \rho_m^n \lambda_{m_{S_w}} (p_m^{n+1} - p_{w_b})(i_G i_W H_m^n + i_G i_T H_m^{n+1})\right]
\end{aligned} \tag{E.37c}$$

$$\begin{aligned}
C_{14} = {} & \frac{V_b}{\Delta t}\left\{\phi^{n+1} \sum_{m=o,w,g} U_m^{n+1} S_m^n \rho_{m_p} + \phi_i c_r \left[\sum_{m=o,w,g} \rho_m^n S_m^n U_m^{n+1} - (\rho C_p)_r T^n\right]\right\} \\
& + \mathrm{PI}\left[\left(\sum_{m=o,w,g} \lambda_m^n \rho_m^n H_m^n\right) i_G i_W + \sum_{m=o,w,g} \lambda_m^n \rho_m^n H_m^{n+1} i_W i_T\right]
\end{aligned} \tag{E.37d}$$

and,

$$\begin{aligned}
R_1 = {} & \Delta\left[\sum_{m=o,w,g} T_m H_m (\Delta p_m - \gamma_m \Delta z)^n\right] \\
& - \frac{V_b}{\Delta t}\phi^{n+1} U_g^{n+1} S_g^{n+1}\left[\rho_g^\ell - \rho_g^n - \rho_{g_T}(T^\ell - T^n) - \rho_{g_p}(p_g^\ell - p_g^n)\right] \\
& + \Delta(T_c \Delta T^n) - \sum_{m=o,w} \rho_m^n H_m^n q_m^n - q_L^n
\end{aligned} \tag{E.38}$$

and,

$$T_{11} = \sum_{m=o,w} (\Delta p_m^{n+1} - \gamma_m \Delta z) T_m^n H_{m_T} i_{T_m} + T_c i_T \tag{E.39a}$$

$$T_{12} = \sum_{m=o,g} H_m^n (\Delta p_m^{n+1} - \gamma_m \Delta z) T \left(\frac{\rho_m}{\mu_m}\right)^n k_{rm S_g} i_{G_m} + H_g^n T_g^n P'_{c_{go}} \tag{E.39b}$$

$$T_{13} = \sum_{m=o,w} H_m^n (\Delta p_m^{n+1} - \gamma_m \Delta z) T \left(\frac{\rho_m}{\mu_m}\right)^n k_{rm S_w} i_{w_m} - H_w^n T_w^n P'_{c_{ow}} \tag{E.39c}$$

$$T_{14} = \sum_{m=o,w,g} H_m^n T_m^n \tag{E.39d}$$

E. 3 *Molar balance for water*

The molar balance for water can be expanded as:

$$\begin{aligned} \frac{V_b}{\Delta t}\delta[\phi(\rho_w S_w + \rho_g S_g)] = \frac{V_b}{\Delta t}\Big\{ & \phi^{n+1} S_g^n [\rho_g^\ell - \rho_g^n - \rho_{g_T}(T^\ell - T^n) - \rho_{g_p}(p_o^\ell - p_o^n)] \\ & + \phi^{n+1}\rho_g^{n+1}\delta S_g + \phi^{n+1}\rho_w^{n+1}\delta S_w + \phi^{n+1} \sum_{m=w,g} S_m^n \rho_{m_T} \delta T \\ & + \left(\phi^{n+1} \sum_{m=w,g} S_m^n \rho_{m_p} + \phi_i c_r \sum_{m=w,g} \rho_m^n S_m^n \right))\delta p \Big\} \end{aligned} \tag{E.40}$$

Writing the difference terms,

$$\begin{aligned} \Delta[T_w(\Delta p_w - \gamma_w \Delta z)]^{n+1} &= \Delta[T_w(\Delta p_w - \gamma_w \Delta z)]^n + \Delta[T_w^n \Delta\delta p_w] \\ &\quad + \Delta[(\Delta p_w - \gamma_w \Delta z)^{n+1} \delta T_w] && \text{(E.41a)} \\ &= \Delta[T_w(\Delta p_w - \gamma_w \Delta z)]^n + \Delta[T_w^n \Delta\delta p_w] - \Delta[T_w^n \Delta(P'_{c_{ow}} \delta S_w)] \\ &\quad + \Delta\left[(\Delta p_w - \gamma_w \Delta z)^{n+1} + \left(\frac{\rho_w}{\mu_w}\right)^n k'_{rw} \delta S_w\right] && \text{(E.41b)} \end{aligned}$$

Similarly,

$$\begin{aligned} \Delta[T_g(\Delta p_g - \gamma_g \Delta z)]^{n+1} = \Delta[T_g(\Delta p_g - \gamma_g \Delta z)]^n + \Delta[T_g^n \Delta\delta p] + \\ \Delta[T_g^n \Delta(P'_{c_{go}} \delta S_g)] + \Delta[(\Delta p_g - \gamma_g \Delta z)]^{n+1} + \Delta\left[\left(\frac{\rho_g}{\mu_g}\right)^n k'_{rw} \delta S_g\right] \end{aligned} \quad \text{(E.41b)}$$

Also,

$$\begin{aligned} (q_w p_w)^{n+1} &= (q_w p_w)^n + \rho_w^n \delta q_w && \text{(E.42a)} \\ &= q_w^n \rho_w^n + \rho_w \mathrm{PI}\left(\frac{k_{rw}^n}{\mu_w^n}\delta S_w (p_w^{n+1} - p_{w_b}) + \frac{k_{rw}^n}{\mu_w^n}\delta p\right) && \text{(E.42b)} \end{aligned}$$

and

$$(q_g p_g)^{n+1} = q_g^n \rho_g^n + \rho_g \mathrm{PI} \left(\frac{k_{rg}^n}{\mu_g^n} \delta S_g (p_g^{n+1} - p_{w_b}) + \frac{k_{rg}^n}{\mu_g^n} \delta p \right) \tag{E.43}$$

Thus the water molar balance equation becomes

$$\begin{aligned}
&\frac{V_b \phi^{n+1}}{\Delta t} \left(\sum_{m=w,g} S_m^n \rho_{m_T} \right) \delta T + \left[\frac{V_b \phi^{n+1}}{\Delta t} \rho_g^{n+1} + \rho_g^n \mathrm{PI} (\rho_g^{n+1} - \rho_{w_b}) \frac{k'_{rg}}{\mu_g^n} \right] \delta S_g \\
&+ \left[\frac{V_b \phi^{n+1}}{\Delta t} \rho_w^{n+1} + \rho_w^n \mathrm{PI} (\rho_w^{n+1} - \rho_{w_b}) \frac{k'_{rw}}{\mu_w^n} \right] \delta S_w \\
&+ \left[\frac{V_b}{\Delta t} \left(\phi^{n+1} \sum_{m=w,g} S_m^n \rho_{m_p} + \phi_i c_r \sum_{m=w,g} \rho_m^n S_m^n \right) + \mathrm{PI} \sum_{m=w,g} \rho_m^n \frac{k_{rm}^n}{\mu_m^n} \right] \delta p \\
&= -\frac{V_b}{\Delta t} \Big\{ \phi^{n+1} \sum_{m=w,g} [\rho_m^\ell - \rho_m^n - \rho_{m_T} (T^\ell - T^n) - p_{m_p} (p_m^\ell - p_m^n)] \\
&\quad + \Delta \left[\sum_{m=w,g} [T_m (\Delta p_m - \gamma_m \Delta z)]^n \right] \\
&\quad - \sum_{m=w,g} q_m^n \rho_m^n + \Delta \left[(\Delta p_g - \gamma_g \Delta z)^{n+1} T \left(\frac{\rho_g}{\mu_g} \right)^n k'_{rg} \delta S_g \right] \\
&\quad + \Delta [T_g^n \Delta (P'_{c_{ow}} \delta S_g)] + \Delta \left[(\Delta p_w - \gamma_w \Delta z)^{n+1} T \left(\frac{\rho_w}{\mu_w} \right)^n k'_{rw} \delta S_w \right] \\
&\quad - \Delta [T_w^n \Delta (P'_{c_{ow}} \delta S_w)] + \Delta \left[\sum_{m=w,g} T_m^n \Delta \delta p \right] \Big\}
\end{aligned} \tag{E.44}$$

Now define the following terms:

$$C_{21} = \frac{V_b \phi^{n+1}}{\Delta t} \sum_{m=w,g} S_w^n \rho_{m_T} \tag{E.45a}$$

$$C_{22} = \frac{V_b \phi^{n+1}}{\Delta t} \rho_g^{n+1} + \mathrm{PI} (p_g^{n+1} - p_{w_b}) \frac{\rho_g^n}{\mu_g^n} k'_{rg} \tag{E.45b}$$

$$C_{23} = \frac{V_b \phi^{n+1}}{\Delta t} \rho_w^{n+1} + \mathrm{PI} (p_w^{n+1} - p_{w_b}) \frac{\rho_w^n}{\mu_w^n} k'_{rw} \tag{E.45c}$$

$$C_{24} = \frac{V_b}{\Delta t} \left(\phi^{n+1} \sum_{m=w,g} S_m^n \rho_{m_p} + \phi_i c_r \sum_{m=w,g} \rho_m^n S_m^n \right) + \mathrm{PI} \sum_{m=w,g} \rho_m^n \frac{k_{rm}^n}{\mu_m^n} \tag{E.45d}$$

and

$$T_{21} = 0 \tag{E.46a}$$

$$T_{22} = (\Delta p_g^{n+1} - \gamma_g^n \Delta z) T \left(\frac{\rho_g}{\mu_g} \right)^n k'_{rg} + T_g^n P'_{c_{go}} \tag{E.46b}$$

$$T_{23} = (\Delta p_w^{n+1} - \gamma_w^n \Delta z) T \left(\frac{\rho_w}{\mu_w} \right)^n k'_{rw} + T_w^n P'_{c_{wo}} \tag{E.46c}$$

$$T_{24} = \sum_{m=w,g} T_m^n \tag{E.46a}$$

and

$$R_2 = -\frac{V_b}{\Delta t}\left\{\phi^{n+1}[S_g^n(\rho_g^\ell - \rho_g^n) - \rho_{g_T}(T^\ell - T^n) - \rho_{g_p}(p^\ell - p^n) + \Delta\left[\sum_{m=w,g} T_m^n(\Delta p_m \gamma_m \Delta z)^n\right] - \sum_{m=w,g} q_m^n \rho_m^n\right\} \quad \text{(E.47)}$$

E. 4 *Molar balance for oil*

Consider each term:

$$\delta(\phi\rho_o S_o) = \phi^{n+1}\rho_o^{n+1}(-\delta S_g - \delta S_w) + \phi^{n+1}S_o^n\rho_{o_T}\delta T + (\phi^{n+1}S_o^n\rho_{o_p} + \rho_o^n S_o^n \phi_i c_r)\delta p \quad \text{(E.48)}$$

Now, expanding,

$$\Delta[T_o(\Delta p_o - \gamma_o \Delta z)]^{n+1} = \Delta[T_o(\Delta p_o - \gamma_o \Delta z)]^n + \Delta[(\Delta p_o - \gamma_o \Delta z)^{n+1}\delta T_o] + \Delta[T_o^n \Delta\delta p] \quad \text{(E.49a)}$$

$$= \Delta[T_o(\Delta p_o - \gamma_o \Delta z)]^n + \Delta[T_o^n \Delta\delta p] + \Delta\left[(\Delta p_o - \gamma_o \Delta z)]^{n+1} + \left(\frac{\rho_o}{\mu_o}\right)^n \left(\frac{\partial k_{ro}}{\partial S_g}\delta S_g + \frac{\partial k_{ro}}{\partial S_w}\delta S_w\right)\right] \quad \text{(E.49b)}$$

and

$$(q_o\rho_o)^{n+1} = q_o^n\rho_o^n + \rho_o \mathrm{PI}\left(\frac{k_{ro}}{\mu_o}\right)^n \delta p + \mathrm{PI}\left(\frac{\rho_o}{\mu_o}\right)^n (p_o^{n+1} - p_{w_b})\left(\frac{\partial k_{ro}}{\partial S_g}\delta S_g + \frac{\partial k_{ro}}{\partial S_w}\delta S_w\right) \quad \text{(E.50)}$$

Thus,

$$C_{31} = \frac{V_b}{\Delta t}\phi^{n+1}S_o^n\rho_{o_T} \quad \text{(E.51a)}$$

$$C_{32} = -\frac{V_b}{\Delta t}\phi^{n+1}\rho_o^{n+1} + \mathrm{PI}\left(\frac{\rho_o}{\mu_o}\right)^n (p_o^{n+1} - p_{w_b})\frac{\partial k_{ro}}{\partial S_g} \quad \text{(E.51b)}$$

$$C_{33} = -\frac{V_b}{\Delta t}\phi^{n+1}\rho_o^{n+1} + \mathrm{PI}\left(\frac{\rho_o}{\mu_o}\right)^n (p_o^{n+1} - p_{w_b})\frac{\partial k_{ro}}{\partial S_w} \quad \text{(E.51c)}$$

$$C_{34} = \frac{V_b}{\Delta t}(\phi^{n+1}S_o^n\rho_{o_p} + \rho_o^n S_o^n \phi_i c_r) + \mathrm{PI}\rho_o^n\frac{k_{ro}^n}{\mu_o^n} \quad \text{(E.51d)}$$

and

$$R_3 = \Delta[T_o(\Delta p_o \gamma_o \Delta z)]^n - q_o^n\rho_o^n \quad \text{(E.52)}$$

E. 5 *Water-steam equilibrium ratio*

This relationship can be discretized as follows:

$$p_g^{n+1} = p_g^\ell + p_{g_T}^\ell(T^{n+1} - T^\ell) \quad \text{(E.53)}$$

and,

$$\delta p_g = p_g^\ell - p_g^n + p_{gT}^\ell (T^{n+1} - T^n + T^n - T^\ell)$$
$$= p_g^\ell - p_g^n + p_{gT}^\ell \delta T + p_{gT}^\ell (T^n - T^\ell) \qquad \text{(E.54)}$$

Now,

$$\delta T = T_{s_p}^\ell \delta p_g - T_{s_p}^\ell (p^\ell - p^n) + T^\ell - T^n \qquad \text{(E.55)}$$

therefore,

$$\delta T - T_{s_p}^\ell \delta p - T_{s_p}^\ell P'_{c_{go}} \delta S_g = T^\ell - T^n - T_{s_p}^\ell (p^\ell - p^n) \qquad \text{(E.56)}$$

whence,

$$C_{41} = 1 \qquad \text{(E.57a)}$$
$$C_{42} = -T_{s_p}^\ell P'_{c_{go}} \qquad \text{(E.57b)}$$
$$C_{43} = 0 \qquad \text{(E.57c)}$$
$$C_{44} = -T_{s_p}^\ell \qquad \text{(E.57d)}$$

and

$$R_4 = T^\ell - T^n - T_{s_p}^\ell (p^\ell - p^n) \qquad \text{(E.58)}$$

The temperature differences can be written as

$$T^{n+1} - T^n = T^\ell - T^n + T_{s_p}(\delta p + P'_{c_{go}} \delta S_g - p_g^\ell + p_g^n) \qquad \text{(E.59)}$$

or,

$$T^{n+1} = T^\ell + T_{s_p}(p_g^n - p_g^\ell + \delta p + P'_{c_{go}} \delta S_g) \qquad \text{(E.60)}$$

but,

$$T^{n+1} = T^\ell + T_{s_p}(p_g^{n+1} - p_g^\ell) \qquad \text{(E.61)}$$

Thus,

$$\delta T = T^\ell - T^n + T_{s_p} \delta p + T_{s_p}(p_g^n - p_g^\ell) \qquad \text{(E.62)}$$

E. 6 *Heat losses to overburden*

Heat losses are calculated using the one-dimensional diffusivity equation in the vertical direction only:

$$\frac{\partial^2 T}{\partial z^2} = \frac{1}{\alpha}\frac{\partial T}{\partial t} \qquad \text{(E.63)}$$

where the thermal diffusivity $\alpha = \frac{\kappa}{\rho c_p}$, and the following initial and boundary conditions apply:

$$T(z, t = 0) = T_o \qquad \text{(E.64a)}$$

$$T(z = 0, t) = T_r(t) \qquad \text{(E.64b)}$$

$$T(z \to \infty, t) = T_o \qquad \text{(E.64c)}$$

The instantaneous heat loss rate is found by integrating and applying the boundary conditions to find

$$q_L(t) = -\kappa A \left.\frac{\partial T}{\partial z}\right|_{z=0} \qquad \text{(E.65)}$$

A suitable solution to this is now found by applying the theorem of superposition. Details can be found in Abou-Kassem (1981).

E.7 *Solution of the equations*

Variable substitution is employed to ensure maximum stability. At all stages three equations are solved for three unknowns. If gas (i.e., steam) and water are present, temperature is no longer an unknown and is replaced by a pressure-gas saturation equation — Equation (E.60). If either the gas or water phase is absent, then either unknown S_g or S_w is replaced by T. Thus, this procedure treats rigorously the cases of gas or water phase disappearance, due to repressurization or vaporization.

The final equations are:

$$\sum_{j=1,4} C_{ij} p_j = R_i + \sum_{j=1,4} \Delta(T_{ij} \Delta p_j), \quad i = 1, 4 \qquad \text{(E.66)}$$

where

$i = 1$ is the energy balance;

$i = 2$ is the molar balance for water;

$i = 3$ is the molar balance for oil;

$i = 4$ is the steam-water equilibrium ratio.

These four equations in four unknowns are reduced to three equations in three unknowns in the following three cases:

(1) $S_g > 0, S_w > 0$;

(2) $S_g = 0, S_w > 0$;

(3) $S_g > 0, S_w = 0$.

For the first case, the energy equation transmissibilities are explicit in temperature, and using Equation (E.60), δT is eliminated from the first equation of the system equations given in Equation (E.66). For the second case, $S_g = 0, \delta S_g = -S_g^n$, and the equilibrium equation does not apply, so that the three unknowns are δT, δS_w, δp.

In matrix notation, this becomes

$$\underset{3\times 3}{\underline{\underline{C}}}\ \underset{3\times 1}{\underline{p}} = \underset{3\times 1}{\underline{R}} + \Delta(\underset{3\times 3}{\underline{\underline{T}}}\ \underset{3\times 1}{\Delta \underline{p}}) \tag{E.67}$$

The system of equations is solved by a direct solution method. The matrices **C**, **T**, and the vector **R** contain $(n+1)$-level terms. Thus iteration is necessary. Before the first iteration, all coefficents are evaluated at the end of the previous time step. New iterates of T^{n+1}, S_w^{n+1}, S_g^{n+1} and p^{n+1} are calculated, and so forth. Convergence criteria are that the max $\delta p < \varepsilon_p$ and max $\delta T < \varepsilon_T$. For most cases two or three iterations are adequate.

E. 8 *Adding solid coupling terms to the formulation*

The new terms are introduced and expanded in Chapter VI. However, the specifics of their incorporation into the finite difference fluid and heat flow model are outlined here. The energy equation has the most terms and will be considered separately

E. 8. 1 *Adding terms to energy equation*

A sequence of modifications can be listed as follows:

(i) Eliminate the original pore compressibility term, $\phi_i c_r$, from Equations (E.15), (E.34), and (E.37), from where it was incorporated into the term C_{14}.

(ii) Add the convective terms contributed from each fluid phase and the rock matrix to R_1.

(iii) Subtract the accumulation terms contributed from each fluid phase and the rock matrix from R_1.

Two of the accumulation terms can be incorporated implicitly by adding Equation (6.25) to term C_{11}. In fact, the first of the two terms, $(1-\phi)\rho_r c_{p_r}$, is part of the original finite difference formulation.

E.8.2 *Adding terms to the flow equations*

Modifications to the two flow equations involve fewer terms, but are otherwise essentially the same as for the energy equation. Redundant pore compressibility terms, $\phi_i c_r$, are removed from the expressions for C_{24} and C_{34}. Convective terms are added to R_2 and R_3, and accumulation terms subtracted from R_2 and R_3.

In all cases (energy and flow equations), if the pore compressibility coupling terms are used, then the corresponding terms replace the original $\phi_i c_r$ terms in the finite difference model, and are not added to the R_1, R_2, and R_3 terms.

Appendix F

EQUATIONS DEFINING SOLID MODEL FOR FINITE ELEMENT ANALYSIS

The essential components of three-dimensional elastoplastic material behaviour are given here, along with the required aspects of geomechanics to be able to formulate the finite element model. The model is based on the two-dimensional computer code published by Owen and Hinton (1982), and has been extended to three-dimensions. Alterations and additions have also been made in the areas of

(i) the residual force calculations;

(ii) the principal stress calculations and the directions;

(iii) the types of loadings permitted;

(iv) the control of the entire program;

(v) the Gauss integration scheme;

(vi) the finite element types.

F. 1 *Geomechanics conventions*

Definitions in continuum and geomechanics are given in this section so that the subsequent sections can be written with an economy of symbols. The definitions are given mostly in index notation at first because of the tensorial nature of the problem. Subsequently, the notation is changed to the vector form commonly used in applied modelling in this area, and all quantities are given expressions which can be evaluated in the vector form.

An essential axiom in continuum mechanics is that stresses are positive in tension, and negative in compression. Thus, in the definition of (compressive) effective stress given by

$$\sigma' = \sigma - u \qquad \text{(F.1)}$$

the pore pressure, u, must be a negative quantity. Thus, to keep pore pressures as positive quantities, the following definition must be used:

$$\sigma' = \sigma + u \qquad \text{(F.2)}$$

The deviatoric stress is defined by

$$s_{ij} = \sigma_{ij} - \delta_{ij}\frac{\sigma_{kk}}{3} \tag{F.3a}$$

$$= \sigma_{ij} - \delta_{ij}\sigma_m \tag{F.3b}$$

where

$$\sigma_m = \frac{\sigma_1 + \sigma_2 + \sigma_3}{3} \tag{F.4}$$

F. 1. 1 *Stress invariants*

(Compiled by Wong, 1989.) Given the characteristic polynomial

$$\sigma^3 - I_1\sigma^2 - I_2\sigma - I_3 = 0 \tag{F.5}$$

The stress invariants I_1, I_2, and I_3, can be found as:

$$I_1 = \sigma_1 + \sigma_2 + \sigma_3 \tag{F.6a}$$

$$= \sigma_x + \sigma_y + \sigma_z \tag{F.6b}$$

$$= 3\sigma_m \tag{F.6c}$$

$$= \sigma_{ii} \tag{F.6d}$$

$$I_2 = -(\sigma_1\sigma_2 + \sigma_2\sigma_3 + \sigma_3\sigma_1) \tag{F.7a}$$

$$= \frac{1}{2}(\sigma_{ij}\sigma_{ji} - \sigma_{ii}\sigma_{jj}) \tag{F.7b}$$

$$I_3 = \frac{1}{3}(\sigma_{ij}\sigma_{jk}\sigma_{ki}) \tag{F.8a}$$

$$= \begin{vmatrix} \sigma_x & \sigma_{xy} & \sigma_{xz} \\ \sigma_{xy} & \sigma_y & \sigma_{yz} \\ \sigma_{xz} & \sigma_{yz} & \sigma_z \end{vmatrix} \tag{F.8b}$$

$$= (\sigma_1\sigma_2\sigma_3) \tag{F.8c}$$

Note that various authors (Mase, 1970, p. 51, Frederick and Chang, 1965, p. 51) use the following form of the characteristic polynomial:

$$\sigma^3 - I_1\sigma^2 + I_2\sigma - I_3 = 0 \tag{F.9}$$

whence the second stress invariant is given by

$$I_2 = \frac{1}{2}(\sigma_{ii}\sigma_{jj} + \sigma_{ij}\sigma_{ji}) \tag{F.10}$$

F. 1. 2 *Deviatoric stress invariants*

The characteristic polynomial in this case is given by

$$s^3 - J_2 s - J_3 = 0 \tag{F.11}$$

The deviatoric stress invariants can be found from this characteristic polynomial. The first deviatoric stress invariant does exist as an analogue of the first stress invariant:

$$J_1 = s_1 + s_2 + s_3 = 0 \tag{F.12}$$

The second deviatoric stress invariant can take many forms:

$$\begin{aligned}
J_2 &= -(s_1 s_2 + s_2 s_3 + s_3 s_1) && \text{(F.13a)}\\
&= \frac{1}{2} s_{ij} s_{ji} && \text{(F.13b)}\\
&= \frac{I_1^2}{3} + I_2 && \text{(F.13c)}\\
&= \frac{1}{2}[(s_x^2 + s_y^2 + s_z^2) + 2(\sigma_{xy}^2 + \sigma_{yz}^2 + \sigma_{zx}^2)] && \text{(F.13d)}\\
&= \frac{1}{2}(s_1^2 + s_2^2 + s_3^2) && \text{(F.13e)}\\
&= \frac{1}{6}\left[(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2\right] && \text{(F.13f)}\\
&= \frac{1}{6}\left[(\sigma_x - \sigma_y)^2 + (\sigma_y - \sigma_z)^2 + (\sigma_z - \sigma_x)^2 + 6(\sigma_{xy}^2 + \sigma_{yz}^2 + \sigma_{zx}^2)\right] && \text{(F.13g)}
\end{aligned}$$

$$\begin{aligned}
J_3 &= \frac{1}{3} s_{ij} s_{jk} s_{ki} && \text{(F.14a)}\\
&= \frac{1}{2}(s_1^3 + s_2^3 + s_3^3) && \text{(F.14b)}\\
&= \sigma_1 \sigma_2 \sigma_3 && \text{(F.14c)}\\
&= \frac{1}{27}(2I_1^3 + 9I_1 I_2 + 27I_3) && \text{(F.14d)}
\end{aligned}$$

F. 1. 3 *Octahedral stress invariants*

(Compiled by Wong, 1989.) These are defined by

$$\sigma_{oct} = \sigma_m = \frac{I_1}{3} \tag{F.15}$$

$$\tau_{oct} = \sqrt{\frac{2}{3}J_2} \tag{F.16}$$

The common soil mechanics parameters p, q, are given by

$$p = \sigma_m = \frac{\sigma_{ii}}{3} = \sigma_{oct} \tag{F.17}$$

$$q = \sigma_q = \frac{3}{\sqrt{2}}\tau_{oct} \tag{F.18}$$

F. 1. 4 *Principal stress relations*

Principal stresses can be found from

$$\sigma_i = \sigma_m - \frac{2}{3}\sigma_q \sin(\theta + \frac{2}{3}(i-2)\pi) \tag{F.19}$$

where

$$\sigma_q = \sqrt{3J_2} = \frac{3}{\sqrt{2}}\tau_{oct} \tag{F.20}$$

The Lode angle, θ, is defined by

$$\tan\theta = \frac{\sigma_1 - 2\sigma_2 + \sigma_3}{\sqrt{3}(\sigma_1 - \sigma_3)} \tag{F.21}$$

or, equivalently, by

$$\sin 3\theta = -\frac{3\sqrt{3}}{2}\frac{J_3}{(J_2)^{\frac{3}{2}}} \tag{F.22}$$

The Lode angle is related to the original Lode parameter, Γ, by

$$\Gamma = -\sqrt{3}\tan\theta \tag{F.23}$$

More principal stress relationships can be found by inspecting the characteristic polynomial for deviatoric stresses:

$$s^3 - J_2 s - J_3 = 0 \tag{F.24}$$

where

$$J_2 = \frac{1}{2} = s_{ij}s_{ji} \tag{F.25}$$

$$J_3 = \frac{1}{3}s_{ij}s_{jk}s_{ki} \tag{F.26}$$

whence, in terms of cosines,

$$s_i = 2\sqrt{\frac{J_2}{3}}\cos\left[\alpha + \frac{2}{3}\pi(i-1)\right] \tag{F.27a}$$

$$= \sqrt{2}\tau_{oct}\cos\left[\alpha + \frac{2}{3}\pi(i-1)\right] \tag{F.27b}$$

where

$$\cos 3\alpha = \frac{\sqrt{2}J_3}{\tau_{oct}^3} \tag{F.28}$$

or, in terms of sines,

$$s_i = 2\sqrt{\frac{J_2}{3}} \begin{cases} \sin(\theta + \frac{2\pi}{3}) \\ \sin(\theta) \\ \sin(\theta + \frac{4\pi}{3}) \end{cases} \tag{F.29}$$

Normal and deviatoric principal stresses are related by

$$\sigma_i = s_i + \frac{I_1}{3} = s_i + \sigma_m \tag{F.30}$$

F.1.5 *Derivatives of stress invariants*

Given (Mase, 1970, p. 188)

$$\frac{\partial \sigma_{ij}}{\partial \sigma_{pq}} = \delta_{ip}\delta_{jq} \tag{F.31}$$

Then, for the first stress invariant (Mase, p28):

$$\frac{\partial I_1}{\partial \underline{\underline{\sigma}}} = \frac{\partial \sigma_{ii}}{\partial \sigma_{pq}} = \delta_{ip}\delta_{iq} = \delta_{pq} \tag{F.32}$$

i.e., the identity matrix $\langle 1, 1, 1, 0, 0, 0 \rangle$. The derivative of the second stress invariant can be found as:

$$\begin{aligned} \frac{\partial I_2}{\partial \underline{\underline{\sigma}}} &= \frac{\partial}{\partial \sigma_{pq}} \left[\frac{1}{2}(\sigma_{ii}\sigma_{jj} - \sigma_{ij}\sigma_{ij}) \right] \\ &= \frac{1}{2} \left[\delta_{ip}\delta_{iq}\sigma_{jj} + \sigma_{ii}\delta_{jp}\delta_{jq} - 2\sigma_{ij}\delta_{ip}\delta_{jq} \right] \\ &= \frac{1}{2} \left[\delta_{pq}(\sigma_{jj} + \sigma_{ii}) - 2\sigma_{pq} \right] \\ &= -(\sigma_{pq} - \delta_{pq}I_1) \equiv -(\sigma_{ij} - \delta_{ij}\sigma_{kk}) \end{aligned} \tag{F.33}$$

The derivative of the third stress invariant is:

$$\begin{aligned} \frac{\partial I_3}{\partial \underline{\underline{\sigma}}} &= \frac{\partial}{\partial \sigma_{pq}} \left[\frac{1}{3}(\sigma_{ij}\sigma_{jk}\sigma_{ki}) \right] \\ &= \frac{1}{3} \left[\delta_{ip}\delta_{jq}\sigma_{jk}\sigma_{ki} + \sigma_{ij}\delta_{jp}\delta_{kq}\sigma_{ki} + \sigma_{ij}\sigma_{jk}\delta_{kp}\delta_{iq} \right] \\ &= \frac{1}{3} \left[\sigma_{qk}\sigma_{kp} + \sigma_{ip}\sigma_{qi} + \sigma_{qj}\sigma_{jp} \right] \\ &= \sigma_{ql}\sigma_{lp} \equiv \sigma_{ik}\sigma_{kj} \end{aligned} \tag{F.34}$$

F. 1. 6 *Derivatives of deviatoric stress invariants*

The derivative of the first deviatoric stress invariant is trivial:

$$\frac{\partial J_1}{\partial \underline{\underline{\sigma}}} = 0 \tag{F.35}$$

Noting that $s_{ii} \equiv J_1 = 0$, then the derivative of the second deviatoric stress invariant is:

$$\begin{aligned}
\frac{\partial J_2}{\partial \underline{\underline{\sigma}}} &= \frac{\partial}{\partial \sigma_{pq}} \left[\frac{1}{2} s_{ij} s_{ij} \right] \\
&= s_{ij} \frac{\partial s_{ij}}{\partial \sigma_{pq}} \\
&= s_{ij} \frac{\partial}{\partial \sigma_{pq}} \left[\sigma_{ij} - \delta_{ij} \frac{\sigma_{kk}}{3} \right] \\
&= s_{ij} \left[\delta_{ip}\delta_{jq} - \delta_{ij} \frac{\delta_{kp}\delta_{kq}}{3} \right] \\
&= s_{pq} - s_{ii} \frac{\delta_{pq}}{3} \\
&= s_{pq} \equiv s_{ij}
\end{aligned} \tag{F.36}$$

The derivative of the third deviatoric stress invariant is:

$$\begin{aligned}
\frac{\partial J_3}{\partial \underline{\underline{\sigma}}} &= \frac{\partial}{\partial \sigma_{pq}} \left[\frac{1}{3} s_{ij} s_{jk} s_{ki} \right] \\
&= \frac{1}{3} \left[\left(\delta_{ip}\delta_{jq} - \frac{\delta_{ij}\delta_{pq}}{3} \right) s_{jk} s_{ki} + s_{ij} \left(\delta_{jp}\delta_{kq} - \frac{\delta_{jk}\delta_{pq}}{3} \right) s_{ki} + s_{ij} s_{jk} \left(\delta_{kp}\delta_{iq} - \frac{\delta_{ki}\delta_{pq}}{3} \right) \right] \\
&= \frac{1}{3} \left[s_{qk} s_{kp} - \frac{\delta_{pq}}{3} s_{ik} s_{ki} + s_{ip} s_{qi} - \frac{\delta_{pq}}{3} s_{ik} s_{ki} + s_{qj} s_{jp} - \frac{\delta_{pq}}{3} s_{kj} s_{jk} \right] \\
&= \frac{1}{3} \left[s_{qk} s_{kp} + s_{ip} s_{qi} + s_{qj} s_{jp} - \frac{3}{3} \delta_{pq} 2 J_2 \right] \\
&= \frac{1}{3} \left[3 s_{qk} s_{kp} - 2 \delta_{pq} J_2 \right] \\
&\equiv s_{ik} s_{kj} - \frac{2}{3} \delta_{ij} J_2
\end{aligned} \tag{F.37}$$

See also Hill (1950) p. 34, Eqn. 22′.

F. 1. 7 *Principal stress values and directions*

The eigenvalues λ and eigenvectors $\underline{x}$ of a matrix $\underline{\underline{A}}$ satisfy

$$\underline{\underline{A}}\,\underline{x} = \lambda \underline{x} \tag{F.38}$$

That is,

$$(\underline{\underline{A}} - \lambda \underline{\underline{I}})\underline{x} = 0 \tag{F.39}$$

If $\underline{x}$ is a basis vector, then $x_1^2 + x_2^2 + x_3^2 = 1$. Consider the vector equation (Frederick and Chang, 1965, p. 54)

$$(\sigma_{ij} - \delta_{ij}\lambda)n_j = 0 \tag{F.40}$$

and the constraint $n_i n_i = 1$. Thus

$$(\sigma_x - \lambda)n_1 + \sigma_{xy}n_2 + \sigma_{xz}n_3 = 0 \tag{F.41}$$

$$\sigma_{xy}n_1 + (\sigma_y - \lambda)n_2 + \sigma_{yz}n_3 = 0 \tag{F.42}$$

$$n_1^2 + n_2^2 + n_3^2 = 1 \tag{F.43}$$

Therefore,

$$n_1 = \frac{-(\sigma_{xy}n_2 + \sigma_{xz}n_3)}{\sigma_x - \lambda} \tag{F.44}$$

From Equation (F.43),

$$\sigma_{xy}\frac{-(\sigma_{xy}n_2 + \sigma_{xz}n_3)}{\sigma_x - \lambda} + (\sigma_y - \lambda)n_2 + \sigma_{yz}n_3 = 0 \tag{F.45}$$

whence

$$n_2 = \frac{a}{b}n_3 \tag{F.46}$$

where

$$a = \sigma_{xy}\sigma_{xz} - (\sigma_x - \lambda)\sigma_{yz}$$

$$b = -\sigma_{xy}^2 - (\sigma_x - \lambda)(\sigma_y - \lambda)$$

Solving finally for n_3 in $n_i n_i = 1$,

$$c^2 n_3^2 + \left(\frac{a}{b}\right)^2 n_3^2 + n_3^2 = 1$$

Thus,

$$n_3^2 = \frac{1}{d} * \text{sign of } \lambda \tag{F.47}$$

where

$$d = c^2 + \left(\frac{a}{b}\right)^2 + 1$$

$$c = \frac{-(\frac{a}{b}\sigma_{xy} + \sigma_{xz})}{\sigma_x - \lambda}$$

whence n_2 and n_1 can be found as

$$n_2 = \frac{a}{b} n_3 \tag{F.48}$$

$$n_1 = \frac{-(\sigma_{xy} n_2 + \sigma_{xz} n_3)}{(\sigma_x - \lambda)} \tag{F.49}$$

This analysis does not define the direction of the vector on the axis (n_1, n_2, n_3), because of the square root evaluation of n_3.

F. 2 *Elastoplastic numerical formulation*

This section follows the approach of Owen and Hinton (1982) for three dimensions, but does not impose the assumption of associated plasticity until the general case has been evaluated. In building the elastoplastic formulation, a few definitions and propositions must be introduced.

F. 2. 1 *Yield criterion*

First, a yield criterion must be proposed:

$$f(\sigma_{ij}) = k(\kappa) \tag{F.50}$$

where f is a function of stress and k is a function of the so-called *hardening parameter* κ. Plastic deformation is considered independent of hydrostatic stress, thus the yield criterion can be written in terms of the deviatoric stress invariants

$$f(J_2, J_3) = k(\kappa) \tag{F.51}$$

where the deviatoric stress $\sigma'_{ij} = \sigma_{ij} - \frac{1}{3}\delta_{ij}\sigma_{kk}$.

F. 2. 2 *Work or strain hardening*

It is postulated that the degree of *work hardening* is a function of the total plastic work, W_p, only.

$$\kappa = W_p \tag{F.52}$$

where

$$W_p = \int \sigma_{ij} \, (d\epsilon_{ij})_p \tag{F.53}$$

One can measure total plastic deformation as the effective, gerneralised or equivalent plastic strain (EPS), defined incrementally in the following tensorial expression:

$$d\bar{\varepsilon}_p = \sqrt{\frac{2}{3}}\left\{(d\varepsilon_{ij})_p\,(d\varepsilon_{ij})_p\right\}^{\frac{1}{2}} \tag{F.54}$$

If the yielding is independent of hydrostatic stress, then $(d\varepsilon_{ii})_p = 0$ and then the plastic strains are all deviatoric strains, i.e., $\left(d\varepsilon'_{ij}\right)_p = (d\varepsilon_{ij})_p$. Thus,

$$d\bar{\varepsilon}_p = \sqrt{\frac{2}{3}}\left\{(d\varepsilon'_{ij})_p\,(d\varepsilon'_{ij})_p\right\}^{\frac{1}{2}} \tag{F.55}$$

Thus one can define the hardening parameter as

$$\kappa = \bar{\varepsilon}_p \tag{F.56}$$

where $\bar{\varepsilon}_p$ is a result of integrating $d\bar{\varepsilon}_p$ over the strain path. This result is known as *strain hardening*. Plastic or elastic material behaviour can now be differentiated by inspection of the value of f:

(i) plastic behaviour is indicated by $f = k$,

(ii) elastic behaviour is indicated by $f < k$,

The result $f > k$ is physically impossible. At the plastic state, $f = k$, the incremental change in the yield function due to incremental stress change is

$$df = \frac{\partial f}{\partial \sigma_{ij}} d\sigma_{ij} \tag{F.57}$$

where

(i) if $df < 0$, there is elastic unloading and the yield point is inside the yield surface;

(ii) if $df = 0$, there is neutral loading, with the stress point on the yield surface, which can be interpreted as plastic loading for a perfectly plastic material, $k = 0$;

(iii) if $df > 0$, there is plastic loading for a strain-hardening material, with the stress point on the yield surface.

F. 2. 3 *The elastoplastic stress-strain relation*

The total strain can be broken into the constituent elastic and plastic strains:

$$d\varepsilon_{ij} = (d\varepsilon_{ij})_e + (d\varepsilon_{ij})_p \tag{F.58}$$

where

$$(d\varepsilon_{ij})_e = \frac{d\sigma'_{ij}}{2\mu} + \frac{(1-2\nu)}{E}\delta_{ij}d\sigma_{kk} \tag{F.59}$$

It is assumed that the plastic strain increment is proportional to the stress gradient of the so-called *plastic potential*, Q, such that

$$(d\varepsilon_{ij})_p = d\lambda\frac{\partial Q}{\partial\sigma_{ij}} \tag{F.60}$$

where $d\lambda$ is a scalar constant of proportionality known as the *plastic multiplier*. Equation (F.60) is known as a *flow rule* because it governs plastic flow after yielding.

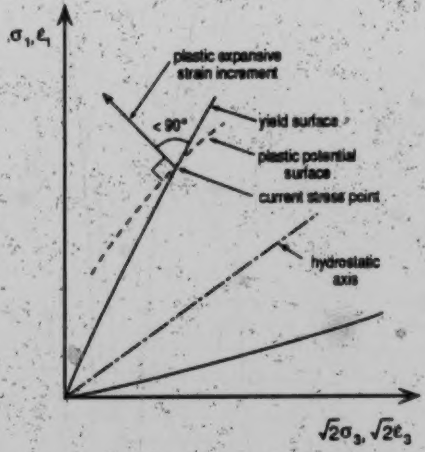

σ_1, ε_1
plastic expansive strain increment
< 90°
yield surface
plastic potential surface
current stress point
hydrostatic axis
$\sqrt{2}\sigma_3, \sqrt{2}\varepsilon_3$

Figure F–1

Yield and plastic potential surfaces for plastic expansive increments

(adapted from Lade and Nelson, 1984)

The plastic potential Q must be a function of the deviatoric stress invariants J_2 and J_3, but is as yet indeterminate in its most general form. It is interpreted graphically in Figure F–1.

It can be postulated that $f \equiv Q$, which is known as the *associated* theory of plasticity. In which case,

$$(d\varepsilon_{ij})_p = d\lambda\frac{\partial f}{\partial\sigma_{ij}} \tag{F.61}$$

This is known also as the *normality condition* because $\frac{\partial f}{\partial\sigma_{ij}}$ is a vector directed normal to the yield surface, at the stress point under consideration.

F.2.4 *Finding the hardening parameter from the uniaxial yield test*

The hardening law $k = k(\kappa)$ can be expressed in terms of effective stress $\bar{\sigma} \propto J_2$. Thus, for a strain hardening material, using Equation (F.56),

$$\bar{\sigma} = H(\bar{\varepsilon}_p) \tag{F.62}$$

The effective stress $\bar{\sigma}$ is analogous to $\bar{\varepsilon}_p$, and is given by

$$\begin{aligned} \bar{\sigma} &= \sqrt{3}\,(J_2)^{\frac{1}{2}} \\ &= \sqrt{\frac{3}{2}}\,(s_{ij}s_{ij})^{\frac{1}{2}} \end{aligned} \tag{F.63}$$

Differentiating Equation (F.62),

$$\frac{d\bar{\sigma}}{d\bar{\varepsilon}_p} = H'(\bar{\varepsilon}_p) \tag{F.64}$$

For the uniaxial case, $\sigma_1 = \sigma$, and $\sigma_2 = \sigma_3 = 0$, thus $\bar{\sigma} = \sigma$. If the plastic strain increment is in the direction of loading and is $d\varepsilon_p$, then $(d\varepsilon_1)_p = d\varepsilon_p$. As plastic straining is assumed to be incompressible, Poisson's ratio = 0.5, whence $(d\varepsilon_2)_p = (d\varepsilon_3)_p = -\frac{1}{2}d\varepsilon_p$. Thus, the effective plastic strain becomes

$$\begin{aligned} d\bar{\varepsilon}_p &= \sqrt{\frac{2}{3}}\left\{(d\varepsilon_{ij})_p\,(d\varepsilon_{ij})_p\right\}^{\frac{1}{2}} \\ &= d\varepsilon_p \end{aligned} \tag{F.65}$$

Thus,

$$\begin{aligned} H'(\bar{\varepsilon}_p) &= \frac{d\sigma}{d\varepsilon_p} \\ &= \frac{d\sigma}{d\varepsilon - d\varepsilon_e} \\ &= \frac{1}{\dfrac{d\varepsilon}{d\sigma} - \dfrac{d\varepsilon_e}{d\sigma}} \\ &= \frac{E_T}{1 - \dfrac{E_T}{E}} \end{aligned} \tag{F.66}$$

where E is the elastic tangent modulus $\dfrac{d\varepsilon_e}{d\sigma}$, and E_T is the tangent modulus at a point, $\dfrac{d\varepsilon}{d\sigma}$. Thus one can obtain H' experimentally from the uniaxial test.

F. 2. 5 *Matrix formulation*

Rewrite the yield function as

$$f(\underline{\sigma}) = k(\kappa) \tag{F.67}$$

From the work hardening hypothesis, Equations (F.52) and (F.53),

$$d\kappa = \underline{\sigma}^T d\underline{\varepsilon}_p \tag{F.68}$$

Alternatively, from the strain-hardening hypothesis, Equation (F.57),

$$d\kappa = d\underline{\varepsilon}_p \tag{F.69}$$

where $\underline{\sigma}$ is the stress vector, and κ is a hardening parameter governing the growth of the yield surface.

Rearranging Equation (F.66),

$$F(\underline{\sigma}, \kappa) = f(\underline{\sigma}) - k(\kappa) = 0 \tag{F.70}$$

Differentiating Equation (F.70), one obtains

$$dF = \frac{\partial F}{\partial \underline{\sigma}} d\underline{\sigma} + \frac{\partial F}{\partial \kappa} d\kappa = 0 \tag{F.71}$$

This can be rewritten as

$$dF = \underline{a}^T d\underline{\sigma} - A d\lambda = 0 \tag{F.72}$$

where

$$\underline{a}^T = \frac{\partial F}{\partial \underline{\sigma}} \tag{F.73}$$

$$A = -\frac{1}{d\lambda} \frac{\partial F}{\partial \kappa} d\kappa \tag{F.74}$$

Here, $\underline{a}$ is known as the *flow vector*. The following three relations can be used to generate the equation for an increment of strain:

$$d\varepsilon_{ij} = (d\varepsilon_{ij})_e + (d\varepsilon_{ij})_p \tag{F.75}$$

$$(d\varepsilon_{ij})_e = \frac{d\sigma'_{ij}}{2\mu} + \frac{(1-2\nu)}{E} \delta_{ij} d\sigma_{kk} \tag{F.76}$$

$$(d\varepsilon_{ij})_p = d\lambda \frac{\partial Q}{\partial \sigma_{ij}} \tag{F.77}$$

Whence,

$$d\varepsilon_{ij} = \frac{d\sigma'_{ij}}{2\mu} + \frac{(1-2\nu)}{E}\delta_{ij}d\sigma_{kk} + d\lambda\frac{\partial Q}{\partial\sigma_{ij}} \tag{F.78}$$

This expression for $d\varepsilon_{ij}$ can be written in vector form as

$$d\underline{\varepsilon} = [\mathrm{D}]^{-1}d\underline{\sigma} + d\lambda\frac{\partial Q}{\partial\underline{\sigma}} \tag{F.79}$$

Premultiply Equation (F.79) by $\underline{a}^T[\mathrm{D}]$ to obtain

$$\underline{a}^T[\mathrm{D}]d\underline{\varepsilon} = \underline{a}^T d\underline{\sigma} + \underline{a}^T[\mathrm{D}]d\lambda\frac{\partial Q}{\partial\underline{\sigma}} \tag{F.80}$$

Now using the result from Equation (F.72), $\underline{a}^T d\underline{\sigma} = Ad\lambda$, this becomes

$$\underline{a}^T[\mathrm{D}]d\underline{\varepsilon} = Ad\lambda + \underline{a}^T[\mathrm{D}]d\lambda\frac{\partial Q}{\partial\underline{\sigma}} \tag{F.81}$$

This can now be rearranged to solve for λ;

$$d\lambda = \frac{1}{A + \underline{a}^T[\mathrm{D}]\underline{b}}\underline{a}^T[\mathrm{D}]d\underline{\varepsilon} \tag{F.82}$$

where

$$\underline{b} = \frac{\partial Q}{\partial\underline{\sigma}} \tag{F.83}$$

Putting Equation (F.82) into Equation (F.79)

$$d\underline{\varepsilon} = [\mathrm{D}]^{-1}d\underline{\sigma} + \frac{1}{A + \underline{a}^T[\mathrm{D}]\underline{b}}\underline{a}^T[\mathrm{D}]d\underline{\varepsilon}\,\underline{b} \tag{F.84}$$

This form can now be solved for $d\underline{\sigma}$:

$$\begin{aligned} d\underline{\sigma} &= \left\{[\mathrm{D}] - \frac{1}{A + \underline{a}^T[\mathrm{D}]\underline{b}}[\mathrm{D}]\underline{b}\,\underline{a}^T[\mathrm{D}]\right\}d\underline{\varepsilon} && \text{(F.85)}\\ &= [\mathrm{D}_{ep}]d\underline{\varepsilon} && \text{(F.86)} \end{aligned}$$

Now, as [D] is symmetric,

$$\underline{a}^T[\mathrm{D}] = [[\mathrm{D}]\underline{a}]^T \tag{F.87}$$

This can be used to define

$$\underline{d}^T = \underline{a}^T[\mathrm{D}] \tag{F.88a}$$

Thus,

$$\underline{d} = [\mathrm{D}]\underline{a} \tag{F.88b}$$

as

$$\underline{d}^T = [[\mathbf{D}]\underline{a}]^T = \underline{a}^T[\mathbf{D}]^T = \underline{a}^T[\mathbf{D}] \tag{F.89}$$

Similarly, define,

$$\underline{d}^{*T} = \underline{b}^T[\mathbf{D}] \tag{F.90a}$$

$$\underline{d}^{*} = [\mathbf{D}]\underline{b} \tag{F.90b}$$

Now, use Equations (F.90) and (F.88) in (F.85) and (F.86), to form

$$[\mathbf{D}_{ep}] = [\mathbf{D}] - \frac{\underline{d}^{*}\underline{d}^T}{A + \underline{d}^T\underline{b}} \tag{F.91}$$

which is the final matrix form of the elastoplastic constitutive matrix.

F. 2. 6 *Determining the value of A*

The work hardening hypothesis can be used in this analysis, which is more general than the strain hardening hypothesis used earlier in this section,

$$d\kappa = \underline{\sigma}^T d\underline{\varepsilon}_p \tag{F.92}$$

A yielding criterion can be obtained from the von Mises postulate that yielding occurs when the second deviatoric stress invariant J_2 reaches a critical value, or

$$(J_2)^{\frac{1}{2}} = k(\kappa) \tag{F.93}$$

but,

$$\underline{\sigma} = \sqrt{3}(J_2)^{\frac{1}{2}} = \sqrt{3}k(\kappa) \tag{F.94}$$

Thus, one can write the uniaxial yield stress

$$\sigma_Y = \sqrt{3}k \tag{F.95}$$

One can also write Equation (F.70) as

$$F(\underline{\sigma}, \kappa) = f(\underline{\sigma}) - \sigma_Y(\kappa) \tag{F.96}$$

Furthermore, using Equation (F.74),

$$\begin{aligned} A &= -\frac{1}{d\lambda}\frac{\partial F}{\partial \kappa}d\kappa \\ &= \frac{1}{d\lambda}\frac{d\sigma_Y}{d\kappa}d\kappa \end{aligned} \tag{F.97}$$

which can be written because $\sigma_Y = f(\kappa)$ only. Rearrange the normality condition in Equation (F.92) to get

$$\begin{aligned} d\kappa &= \underline{\sigma}^T d\underline{\varepsilon}_p \\ &= \underline{\sigma}^T d\lambda \underline{b} \\ &= d\lambda \underline{b}^T \underline{\sigma} \end{aligned} \tag{F.98}$$

where $\underline{b} = \dfrac{\partial Q}{\partial \underline{\sigma}}$. From Equation (F.64), one obtains

$$\frac{d\bar{\sigma}}{d\bar{\varepsilon}_p} = \frac{d\sigma_Y}{d\bar{\varepsilon}_p} = H' \tag{F.99}$$

But, for the uniaxial case, the effective stress and strain are given by $\underline{\sigma} = \bar{\sigma} = \sigma_Y$ and $d\underline{\varepsilon}_p = d\bar{\varepsilon}_p$. Thus,

$$\begin{aligned} d\kappa &= \sigma_Y d\bar{\varepsilon}_p \\ &= d\lambda \underline{b}^T \underline{\sigma} \end{aligned} \tag{F.100}$$

Euler's theorem on homogenous functions states that if $f(\underline{x})$ is homogenous and of degree n, then

$$\frac{\partial f}{\partial \underline{x}} \cdot \underline{x} = nf \tag{F.101}$$

Thus, from Equation (F.96),

$$\frac{\partial f}{\partial \underline{\sigma}} \underline{\sigma} = \sigma_Y = \underline{a}^T \underline{\sigma} \tag{F.102}$$

Comparing this result to Equation (F.100),

$$\sigma_Y d\bar{\varepsilon}_p = d\lambda \underline{b}^T \underline{\sigma} \tag{F.100}$$

For associated plasticity, $\underline{a} = \underline{b}$, which gives for the last two results,

$$\sigma_Y d\bar{\varepsilon}_p = d\lambda \sigma_Y \tag{F.103}$$

which will lead directly to the equivalence of A and H'. Alternatively, from Equation (F.97),

$$\begin{aligned} A &= \frac{1}{d\lambda} \frac{\partial \sigma_Y}{\partial \kappa} d\kappa \\ &= \frac{d\sigma_Y}{d\lambda} \end{aligned} \tag{F.104}$$

and from Equation (F.100),

$$d\kappa = d\lambda \underline{b}^T \underline{\sigma} \tag{F.99}$$

These results are used in Equation (F.104) to obtain

$$\begin{aligned} A &= \frac{1}{d\lambda}\frac{d\sigma_Y}{d\kappa}d\lambda\underline{b}^T\underline{\sigma} \\ &= \frac{d\sigma_Y}{d\kappa}\underline{b}^T\underline{\sigma} \end{aligned} \tag{F.105}$$

From the work hardening hypothesis, Equation (F.92),

$$\begin{aligned} d\kappa &= \sigma_Y d\bar{\varepsilon}_p \\ &= d\lambda\underline{b}^T\underline{\sigma} \end{aligned} \tag{F.106}$$

Rearranging, one obtains

$$d\bar{\varepsilon}_p = \frac{d\lambda\underline{b}^T\underline{\sigma}}{\sigma_Y} \tag{F.107}$$

If Equation (F.105), is solved for A, using Equation (F.106),

$$\begin{aligned} A &= \frac{d\sigma_Y}{d\kappa}\underline{b}^T\underline{\sigma} \\ &= \frac{d\sigma_Y}{\sigma_Y d\bar{\varepsilon}_p}\underline{b}^T\underline{\sigma} \\ &= \frac{d\sigma_Y}{d\bar{\varepsilon}_p}\frac{\underline{b}^T\underline{\sigma}}{\sigma_Y} \end{aligned} \tag{F.108}$$

But, from (F.64), for uniaxial stress,

$$\begin{aligned} H' &= \frac{d\bar{\sigma}}{d\bar{\varepsilon}_p} \\ &= \frac{d\sigma_Y}{d\bar{\varepsilon}_p} \end{aligned} \tag{F.109}$$

Whence, the relation between A and H',

$$A = H'\frac{\underline{b}^T\sigma}{\sigma_Y} \tag{F.110}$$

If associated plasticity is assumed, then $\underline{b} = \underline{a}$, and, using Equation (F.102)

$$\frac{\underline{a}^T\underline{\sigma}}{\sigma_Y} = 1 \tag{F.111}$$

whence, for associated plasticity,

$$A = H' \tag{F.112}$$

Thus, only the plastic strain derivative of the yield stress is required to incorporate the hardening parameter into the matrix formulation of elastoplasticity. If needed, Equation (F.110) provides the general relationship for non-associated plasticity.

F. 3 *Evaluation of key quantities in elastoplastic formulation*

The preceding section outlined the elastoplastic numerical formulation. Many of the quantities were referred to in general algebraic terms. These will now be explained in quantifiable terms that can be implemented in a computer code.

F. 3. 1 *Elastoplastic matrix*

The standard form of the elastoplastic stiffness matrix is given by

$$\frac{\mathbf{D}^E : \dfrac{\partial G}{\partial \underline{\sigma}} : \dfrac{\partial F}{\partial \underline{\sigma}} : \mathbf{D}^E}{H + \dfrac{\partial F}{\partial \underline{\sigma}} : \mathbf{D}^E : \dfrac{\partial G}{\partial \underline{\sigma}}} \tag{F.113}$$

or, in the more familiar index notation,

$$\frac{D^E_{ijmn} \dfrac{\partial G}{\partial \sigma_{mn}} : D^E_{klrs} \dfrac{\partial F}{\partial \sigma_{rs}}}{H + \dfrac{\partial F}{\partial \sigma_{ij}} D^E_{ijkl} \dfrac{\partial G}{\partial \sigma_{kl}}} \tag{F.114}$$

Write the stress tensor as a vector of six components because of the symmetry of the stress tensor.

$$\underline{\sigma} = \{\sigma_x, \sigma_y, \sigma_z, \tau_{xy}, \tau_{yz}, \tau_{zx}\} \tag{F.115a}$$

$$= \{\sigma_{11}, \sigma_{22}, \sigma_{33}, \sigma_{23}, \sigma_{31}, \sigma_{12}\} \tag{F.115b}$$

The yield function used in this work was that of the Mohr-Coulomb model. This model is given in terms of the stress invariants I_1, J_2, and θ, where

$$\theta = -\frac{1}{3}\sin^{-1}\left(-\frac{3\sqrt{3}}{2}\frac{J_3}{J_2^{\frac{3}{2}}}\right) \tag{F.116}$$

The Mohr-Coulomb yield function can be written in terms of the stress invariants as:

$$\left.\begin{aligned} F &= \frac{1}{3} I_1 \sin\phi + (J_2)^{\frac{1}{2}} \left(\cos\theta - \frac{1}{\sqrt{3}}\sin\theta\sin\phi\right) \\ &= c\cos\phi \end{aligned}\right\} \tag{F.117}$$

F. 3. 2 *Derivatives of yield function with respect to $\underline{\sigma}$*

Expand the derivative of the yield function with respect to the stress vector in terms of derivatives with respect to I_1, $(J_2)^{\frac{1}{2}}$, and θ:

$$\begin{aligned}\underline{a}^T &= \frac{\partial F}{\partial \underline{\sigma}}\\ &= \frac{\partial F}{\partial I_1}\frac{\partial I_1}{\partial \underline{\sigma}} + \frac{\partial F}{\partial (J_2)^{\frac{1}{2}}}\frac{\partial (J_2)^{\frac{1}{2}}}{\partial \underline{\sigma}} + \frac{\partial F}{\partial \theta}\frac{\partial \theta}{\partial \underline{\sigma}}\end{aligned} \tag{F.118}$$

Some of these derivatives have already been derived in their tensor form. It will be necessary to convert them to an (x, y, z) form. Start with $\frac{\partial}{\partial \theta}\underline{\sigma}$

$$\begin{aligned}\frac{\partial \theta}{\partial \underline{\sigma}} &= \frac{\partial \theta}{\partial (\sin 3\theta)}\frac{\partial (\sin 3\theta)}{\partial \underline{\sigma}}\\ &= \frac{1}{3\cos 3\theta}\frac{\partial (\sin 3\theta)}{\partial \underline{\sigma}}\end{aligned} \tag{F.119}$$

This is further expanded as

$$\frac{\partial (\sin 3\theta)}{\partial \underline{\sigma}} = \frac{\partial (\sin 3\theta)}{\partial (J_2)^{\frac{1}{2}}}\frac{\partial (J_2)^{\frac{1}{2}}}{\partial \underline{\sigma}} + \frac{\partial (\sin 3\theta)}{\partial (J_3)}\frac{\partial (J_3)}{\partial \underline{\sigma}} \tag{F.120}$$

Referring to the equation for θ as a stress invariant,

$$\sin 3\theta = -\frac{3\sqrt{3}}{2}\frac{J_3}{(J_2)^{\frac{3}{2}}} \tag{F.121}$$

Differentiating with respect to $(J_2)^{\frac{1}{2}}$,

$$\begin{aligned}\frac{\partial (\sin 3\theta)}{\partial (J_2)^{\frac{1}{2}}} &= -\frac{3\sqrt{3}}{2}\frac{-3J_3}{(J_2)^{\frac{4}{2}}}\\ &= \frac{9\sqrt{3}}{2}\frac{J_3}{(J_2)^2}\end{aligned} \tag{F.122}$$

and differentiating with respect to J_3,

$$\frac{\partial (\sin 3\theta)}{\partial (J_3)} = -\frac{3\sqrt{3}}{\text{[illegible]}}\left(\frac{1}{(J_2)^{\frac{3}{2}}}\right) \tag{F.123}$$

whence

$$\frac{\partial \theta}{\partial \underline{\sigma}} = \frac{1}{3\cos 3\theta}\frac{-3\sqrt{3}}{2}\left[\frac{1}{(J_2)^{\frac{3}{2}}}\frac{\partial (J_3)}{\partial \underline{\sigma}} - \frac{3J_3}{(J_2)^2}\frac{\partial (J_2)^{\frac{1}{2}}}{\partial \underline{\sigma}}\right] \tag{F.124}$$

Other terms in (F.118) are

$$\frac{\partial F}{\partial I_1} = \frac{1}{3}\sin \phi \tag{F.125}$$

$$\frac{\partial F}{\partial (J_2)^{\frac{1}{2}}} = \cos \theta (1 - \frac{1}{\sqrt{3}}\tan \theta \sin \phi) \tag{F.126}$$

Now, break down the flow vector $\underline{a}$ into the components

$$\underline{a} = C_1\underline{a}_1 + C_2\underline{a}_2 + C_3\underline{a}_3 \qquad \text{(F.127)}$$

where

$$\underline{a}_1^T = \frac{\partial I_1}{\partial \underline{\sigma}} = \langle 1, 1, 1, 0, 0, 0\rangle \qquad \text{(F.128)}$$

as was found earlier. The second component is given by

$$\begin{aligned} \underline{a}_2^T &= \frac{\partial (J_2)^{\frac{1}{2}}}{\partial \underline{\sigma}} \\ &= \frac{\partial (J_2)^{\frac{1}{2}}}{\partial J_2}\frac{\partial J_2}{\partial \underline{\sigma}} \\ &= \frac{1}{2(J_2)^{\frac{1}{2}}}\frac{\partial J_2}{\partial \underline{\sigma}} \end{aligned} \qquad \text{(F.129)}$$

Therefore,

$$\underline{a}_2^T = \frac{1}{2(J_2)^{\frac{1}{2}}}\langle s_x, s_y, s_z, 2s_{yz}, 2s_{zx}, 2s_{xy}\rangle \qquad \text{(F.130)}$$

Finally,

$$\underline{a}_3^T = \frac{\partial J_3}{\partial \underline{\sigma}} \qquad \text{(F.131)}$$

This can be expanded by using the tensor value of the derivative and manipulating the result

$$\frac{\partial J_3}{\partial \underline{\sigma}} = s_{ik}s_{kj} - \frac{2}{3}\delta_{ij}J_2 \qquad \text{(F.132)}$$

Expand the terms for $\sigma_x \equiv \sigma_{11}$, $\sigma_y \equiv \sigma_{22}$, and $\sigma_z \equiv \sigma_{33}$, by subtracting J_2 expanded, adding J_2 literally, and adding $\frac{1}{2}J_1^2 (= 0)$. Expand the terms for $\sigma_{yz} \equiv \sigma_{23}$, $\sigma_{zx} \equiv \sigma_{31}$, and $\sigma_{yx} \equiv \sigma_{12}$, by adding and subtracting $s_x\sigma_{yz}$, $s_y\sigma_{zx}$, $s_z\sigma_{xy}$ respectively to the terms, and noting that $J_1 = s_x + s_y + s_z = 0$. Whence, one can obtain,

$$\begin{aligned} \frac{\partial J_3}{\partial \underline{\sigma}} = \Big\{ &\left(s_y s_z - \tau_{yz}^2 + \frac{J_2}{3}\right), \left(s_z s_x - \tau_{zx}^2 + \frac{J_2}{3}\right), \left(s_x s_y - \tau_{xy}^2 + \frac{J_2}{3}\right), \\ &2(\tau_{xy}\tau_{zx} - s_x\tau_{yz}), 2(\tau_{xy}\tau_{yz} - s_y\tau_{zx}), 2(\tau_{zx}\tau_{yz} - s_z\tau_{yx})\Big\} \end{aligned} \qquad \text{(F.133)}$$

The scalar coefficients are given by:

$$C_1 = \frac{\partial F}{\partial I_1} = \frac{1}{3}\sin\phi \qquad \text{(F.134)}$$

and

$$
\begin{aligned}
C_2 &= \frac{\partial F(J_2,\theta)}{\partial (J_2)^{\frac{1}{2}}} \\
&= \frac{\partial F(J_2)}{\partial (J_2)^{\frac{1}{2}}} + \frac{\partial F(\theta)}{\partial \theta}\frac{\partial \theta}{\partial (J_2)^{\frac{1}{2}}} \\
&= \frac{\partial F(J_2)}{\partial (J_2)^{\frac{1}{2}}} - \frac{\tan 3\theta}{(J_2)^{\frac{1}{2}}}\frac{\partial F(\theta)}{\partial \theta}
\end{aligned}
\tag{F.135}
$$

where,

$$
\frac{\partial F(\theta)}{\partial \theta} = (J_2)^{\frac{1}{2}}\left(-\sin\theta - \frac{1}{\sqrt{3}}\cos\theta\sin\phi\right)
\tag{F.136}
$$

and

$$
\begin{aligned}
\frac{\partial \theta}{\partial (J_2)^{\frac{1}{2}}} &= \frac{1}{3\cos 3\theta}\frac{\partial(\sin 3\theta)}{\partial (J_2)^{\frac{1}{2}}} \\
&= \frac{1}{3\cos 3\theta}\frac{9\sqrt{3}}{2}\frac{J_3}{(J_2)^2} \\
&= \frac{3\sqrt{3}}{2}\frac{J_3}{(J_2)^{\frac{3}{2}}}\frac{1}{(J_2)^{\frac{1}{2}}}\frac{1}{\cos 3\theta} \\
&= -\frac{\sin 3\theta}{\cos 3\theta}\frac{1}{(J_2)^{\frac{1}{2}}} \\
&= -\frac{\tan 3\theta}{(J_2)^{\frac{1}{2}}}
\end{aligned}
\tag{F.137}
$$

whence,

$$
\begin{aligned}
C_2 &= \cos\theta\left(1 - \frac{1}{\sqrt{3}}\cos\theta\sin\phi\right) - (J_2)^{\frac{1}{2}}\left(-\sin\theta - \frac{1}{\sqrt{3}}\cos\theta\sin\phi\right)\frac{\tan 3\theta}{(J_2)^{\frac{1}{2}}} \\
&= \cos\theta\left[(1 + \tan\theta\tan 3\theta) + \sin\phi\frac{(\tan 3\theta - \tan\theta)}{\sqrt{3}}\right]
\end{aligned}
\tag{F.138}
$$

The last scalar coefficient is given by

$$
C_3 = \frac{\partial F(\theta)}{\partial \theta}\frac{\partial \theta}{\partial (J_3)}
\tag{F.139}
$$

where

$$
\begin{aligned}
\frac{\partial \theta}{\partial (J_3)} &= \frac{1}{3\cos 3\theta}\frac{\partial(\sin 3\theta)}{\partial J_3} \\
&= \frac{1}{3\cos 3\theta}\frac{-3\sqrt{3}}{2}\frac{1}{(J_2)^{\frac{3}{2}}} \\
&= -\frac{\sqrt{3}}{2(J_2)^{\frac{3}{2}}}\frac{1}{\cos 3\theta}
\end{aligned}
\tag{F.140}
$$

Whence

$$C_3 = (J_2)^{\frac{1}{2}}\left(-\sin\theta - \frac{1}{\sqrt{3}}\cos\theta\sin\phi\right)\frac{-\sqrt{3}}{2(J_2)^{\frac{3}{2}}}\frac{1}{\cos 3\theta}$$

$$= \frac{\sqrt{3}\sin\theta + \cos\theta\sin\phi}{2J_2\cos 3\theta} \qquad \text{(F.141)}$$

F. 3. 3 *Elastic constitutive matrix* $\mathbf{D}_e$

The elastic constitutive matrix relates stress to strain as

$$\sigma_K = D_{KM}\epsilon_M \qquad \text{(F.142)}$$

For an isotropic elastic medium, the elastic constitutive matrix D is given by (e.g., Mase, 1970),

$$D_{KM} = \begin{bmatrix} (\lambda+2\mu) & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & (\lambda+2\mu) & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & (\lambda+2\mu) & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu \end{bmatrix} \qquad \text{(F.143)}$$

The elastic stress-strain relation can alternatively be written in the tensor form

$$\sigma_{ij} = \lambda\delta_{ij}\varepsilon_{kk} + 2\mu\varepsilon_{ij} \qquad \text{(F.144)}$$

and the Lamé constant λ and the shear modulus $\mu \equiv G$ are given by

$$\mu = \frac{E}{2(1+\nu)} \qquad \text{(F.145a)}$$

$$\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)} = \frac{2\mu\nu}{(1-2\nu)} \qquad \text{(F.145b)}$$

F. 3. 4 *Strain-displacement matrix*

The strain vector is related to the displacements through the strain-displacement matrix, L, (e.g., Smith and Griffiths, 1988),

$$\underline{\varepsilon} = \underline{\underline{L}}\,\underline{u} \qquad \text{(F.146)}$$

which can be fully written as

$$\begin{Bmatrix} \epsilon_x \\ \epsilon_y \\ \epsilon_z \\ \gamma_{yz} \\ \gamma_{zx} \\ \gamma_{xy} \end{Bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 \\ 0 & \frac{\partial}{\partial y} & 0 \\ 0 & 0 & \frac{\partial}{\partial z} \\ 0 & \frac{\partial}{\partial z} & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} & 0 \end{bmatrix} \begin{Bmatrix} u \\ v \\ w \end{Bmatrix} \tag{F.147}$$

Furthermore, the **B** matrix is related to **L** as

$$\underline{\underline{B}} = \underline{\underline{L}} N_i \tag{F.148}$$

Thus, for three degrees of freedom and 8-noded elements, **B** is a 6 × 24 matrix.

F. 3. 5 *Determinants using stresses in vector form*

The third deviatoric stress invariant is given in index notation by

$$J_3 = \frac{1}{3} s_{ij} s_{jk} s_{ki} \tag{F.149}$$

This can be expanded as the determinant

$$\begin{aligned} \frac{1}{3} s_{ij} s_{jk} s_{ki} &= \begin{vmatrix} s_{11} & s_{12} & s_{13} \\ s_{21} & s_{22} & s_{23} \\ s_{31} & s_{32} & s_{33} \end{vmatrix} \\ &= s_{11}(s_{22}s_{33} - s_{23}s_{32}) - s_{12}(s_{21}s_{33} - s_{31}s_{23}) + s_{13}(s_{21}s_{32} - s_{22}s_{31}) \end{aligned} \tag{F.150}$$

Writing the deviatoric stresses in the following vector form, noting the symmetry of the stress tensor,

$$\underline{s} = \langle s_{11}, s_{22}, s_{33}, s_{23}, s_{31}, s_{12} \rangle \tag{F.151}$$

the value of J_3 can be found in terms of the vector entries,

$$\begin{aligned} \frac{1}{3} s_{ij} s_{jk} s_{ki} &= s(1)[s(2)s(3) - s(5)s(4)] \\ &\quad - s(6)[s(6)s(3) - s(5)s(4)] + s(5)[s(6)s(4) - s(2)s(5)] \end{aligned} \tag{F.152}$$

F. 3. 6 *Evaluation of effective plastic strain*

The current yield stress can be found from

$$\sigma_Y = H' d\bar{\varepsilon}_p \tag{F.153}$$

Normally, the $d\bar{\varepsilon}_p$ term is found from the expression in Equation (F.107),

$$d\bar{\varepsilon}_p = \frac{d\lambda \underline{b}^T \underline{\sigma}}{\sigma_Y} \tag{F.107}$$

However, for a cohesionless material ($c = 0$), and the Mohr-Coulomb failure criterion in explicit form:

$$(\sigma_1 - \sigma_3) = 2c\cos\phi - (\sigma_1 + \sigma_3)\sin\phi \tag{F.154}$$

material failure will occur at zero effective stress $\bar{\sigma}$. For this set of conditions, then, Equation (F.107) cannot be evaluated. Instead, it is necessary to evaluate $d\bar{\varepsilon}_p$ explicitly, viz. expand the tensor expression defining it:

$$d\bar{\varepsilon}_p = \sqrt{\frac{2}{3}}\left\{\left(d\varepsilon'_{ij}\right)_p \left(d\varepsilon'_{ij}\right)_p\right\}^{\frac{1}{2}} \tag{F.155}$$

To evaluate this, the tensor product $\varepsilon_{ij}\varepsilon_{ij}$ will be expanded, making use of the symmetric nature of the strain tensor

$$\varepsilon_{ij}\varepsilon_{ij} = (\varepsilon_{11}^2 + \varepsilon_{22}^2 + \varepsilon_{33}^2) + 2(\varepsilon_{12}^2 + \varepsilon_{23}^2 + \varepsilon_{31}^2) \tag{F.156}$$

By definition,

$$\begin{aligned}\left(d\varepsilon_{ij}\right)_p &= d\lambda \frac{\partial f}{\partial \sigma_{ij}} \\ &= d\lambda \underline{a}\end{aligned} \tag{F.157}$$

Thus, using the expansion in Equation (F.156), and the equivalence between the tensor and vector forms shown earlier,

$$d\bar{\varepsilon}_p = \sqrt{\frac{2}{3}} d\lambda \sqrt{a_1^2 + a_2^2 + a_3^2 + 2(a_4^2 + a_5^2 + a_6^2)} \tag{F.158}$$

F. 4 *Incorporating loads into the formulation*

Loads can be self-weight (gravity), pore pressure changes, temperature changes, and constant stress loads. They are incorporated as nodal forces, applied so that the finite element model can equilibrate the forces in the structure. The convention in soil mechanics of u for pore pressure and F for loads will be used in this section.

F.4.1 *Gravity loadings*

Let gravity always be acting in the negative z-direction, i.e., (Hinton and Owen, 1977, p. 146):

$$\vec{g} = (0, 0, -g) \tag{F.159}$$

The force due to gravity, dG, acting on an elemental volume $d\Omega$ is

$$dG = \rho g d\Omega \tag{F.160}$$

Breaking this down into the directional components:

$$\left.\begin{aligned} dG_x &= 0 \\ dG_y &= 0 \\ dG_z &= -\rho g d\Omega \end{aligned}\right\} \tag{F.161}$$

One can use the principal of virtual work and apply the virtual displacements u^*, v^*, w^* respectively in the x, y, z directions. Whence the equivalent nodal forces are:

$$\left.\begin{aligned} F_{x_i} u^* &= 0 \\ F_{y_i} v^* &= 0 \\ F_{z_i} w^* &= \int_{\Omega^e} -N_i w^* \rho g \, d\Omega \end{aligned}\right\} \tag{F.162}$$

Using the principle of virtual work, this statement holds true for all u^*, v^*, w^*, whence, assuming ρ is constant in the element,

$$\begin{Bmatrix} F_{x_i} \\ F_{y_i} \\ F_{z_i} \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ \int_{\Omega^e} -(\rho g) N_i \, d\Omega \end{Bmatrix} \tag{F.163}$$

To incorporate a spatial variation in density, consider the principle of virtual work applied to a load at a point ξ_p in a one-dimensional system, which yields

$$F_{z_i} w^* = F_z N_i(\xi_p) w^* \tag{F.164}$$

or, simply

$$F_{z_i} = F_z N_i(\xi_p) \tag{F.165}$$

For gravity loading, the increment of load

$$\begin{aligned} dF_{z_i} &= dF_z N_i(\xi_p) \\ &= -(\rho g) d\Omega N_i(\xi_p) \end{aligned} \tag{F.166}$$

Integrating the work expression,

$$\begin{aligned} \int_{\Omega^e} dF_{z_i} w^* &= F_{z_i} w^* \\ &= \int_{\Omega^e} -(\rho g) N_i w^* \, d\Omega \\ &= \int_{\Omega^e} (-g)(\rho_j N_j) N_i w^* \, d\Omega \end{aligned} \tag{F.167}$$

where the following approximation for ρ was used:

$$\rho \approx \rho_j N_j \tag{F.168}$$

Thus the load vector for a system of spatially varying density is

$$\begin{Bmatrix} F_{x_i} \\ F_{y_i} \\ F_{z_i} \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ -\int_{\Omega^e} g \rho_j N_j N_i \, d\Omega \end{Bmatrix} \tag{F.169}$$

The product $\rho_j N_j$ will therefore have to be found at each quadrature point.

F. 4. 2 *Pore pressure loads*

Changes in pore pressure are treated as an external loading and are considered to be a distributed body force with local intensity (force/unit volume) (Naylor, 1978)

$$p = -\nabla(\Delta u) \tag{F.170}$$

Using the same principle of virtual work as for gravity loading, one obtains

$$F_{i_x} = -\int_{\Omega^e} \frac{\partial N_i}{\partial x} \Delta u \, d\Omega \tag{F.171}$$

Approximate the pore pressure distribution using trial functions,

$$\Delta u \approx N_j \Delta u_j \tag{F.172}$$

whence the load vector due to changes in pore pressure:

$$\begin{Bmatrix} F_{x_i} \\ F_{y_i} \\ F_{z_i} \end{Bmatrix} = \begin{Bmatrix} -\int_{\Omega^e} \frac{\partial N_i}{\partial x}(N_j \Delta u_j)\, d\Omega \\ -\int_{\Omega^e} \frac{\partial N_i}{\partial y}(N_j \Delta u_j)\, d\Omega \\ -\int_{\Omega^e} \frac{\partial N_i}{\partial z}(N_j \Delta u_j)\, d\Omega \end{Bmatrix} \tag{F.173}$$

F.4.3 *Thermal loads*

Changes in temperature are accommodated by the use of equilibrium equations obtained through the principle of virtual work, in particular (Hinton and Owen, 1977, p. 18)

$$F^e_{\sigma^0} = -\int_{\Omega^e} [\mathrm{B}]^T \underline{\sigma}^0\, d\Omega \tag{F.174}$$

and

$$\underline{\sigma} = \underline{\underline{D}}\left(\underline{\underline{B}}\,\underline{\delta}^e - \underline{\varepsilon}^0\right) + \underline{\sigma}^0) \tag{F.175}$$

The initial strains are given by (Timoshenko and Goodier, 1982, p. 456)

$$\varepsilon^0_i = \alpha \Delta T \tag{F.176}$$

while the angular distortions

$$\gamma^0_{ij_{i \neq j}} = 0 \tag{F.177}$$

because free thermal expansion does not produce angular distortion in an isotropic material. The initial stress corresponding to these strains are now evaluated from

$$\sigma^0_i = D_{ij}\varepsilon^0_j \tag{F.178}$$

These inital stresses can now be converted into equivalent nodal loads through Equation (0.173):

$$\begin{Bmatrix} F_{x_i} \\ F_{y_i} \\ F_{z_i} \end{Bmatrix} = -\int_{\Omega^e} [\mathrm{B}_i]^T \underline{\sigma}^0\, d\Omega \tag{F.179}$$

The change in temperature must be found at the quadrature points to integrate this expression. Thus the approximation

$$\Delta T \approx N_j T_j \tag{F.180}$$

is used. $[\mathbf{B}_i]$ is the matrix of derivatives of the shape functions, viz.,

$$[\mathbf{B}_i] = \begin{bmatrix} \frac{\partial N_i}{\partial x} & 0 & 0 \\ 0 & \frac{\partial N_i}{\partial y} & 0 \\ 0 & 0 & \frac{\partial N_i}{\partial z} \\ 0 & \frac{\partial N_i}{\partial z} & \frac{\partial N_i}{\partial y} \\ \frac{\partial N_i}{\partial z} & 0 & \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} & \frac{\partial N_i}{\partial x} & 0 \end{bmatrix} \tag{F.181}$$

The transpose of $[\mathbf{B}_i]$, $[\mathbf{B}_i]^T$, is therefore,

$$[\mathbf{B}_i]^T = \begin{bmatrix} \frac{\partial N_i}{\partial x} & 0 & 0 & 0 & \frac{\partial N_i}{\partial z} & \frac{\partial N_i}{\partial y} \\ 0 & \frac{\partial N_i}{\partial y} & 0 & \frac{\partial N_i}{\partial z} & 0 & \frac{\partial N_i}{\partial x} \\ 0 & 0 & \frac{\partial N_i}{\partial z} & \frac{\partial N_i}{\partial y} & \frac{\partial N_i}{\partial x} & 0 \end{bmatrix} \tag{F.182}$$

The initial stress vector is given by

$$\underline{\sigma}^0 = \langle \sigma_x^0, \sigma_y^0, \sigma_z^0, \tau_{yz}^0, \tau_{zx}^0, \tau_{xy}^0 \rangle^T \tag{F.183}$$

Thus the product $[\mathbf{B}_i]^T \underline{\sigma}^0$ is given by

$$[\mathbf{B}_i]^T \underline{\sigma}^0 = \begin{bmatrix} \frac{\partial N_i}{\partial x}\sigma_x^0 + \frac{\partial N_i}{\partial z}\tau_{xz}^0 + \frac{\partial N_i}{\partial y}\tau_{xy}^0 \\ \frac{\partial N_i}{\partial y}\sigma_y^0 + \frac{\partial N_i}{\partial z}\tau_{yz}^0 + \frac{\partial N_i}{\partial x}\tau_{xy}^0 \\ \frac{\partial N_i}{\partial z}\sigma_z^0 + \frac{\partial N_i}{\partial y}\tau_{yz}^0 + \frac{\partial N_i}{\partial x}\tau_{xz}^0 \end{bmatrix} \tag{F.184}$$

F.4.4 *Constant stress loading*

This is a simple case of distributed loads, described for two dimensions in Hinton and Owen (1977), p. 150 and p. 18. These loads are the type imposed by the regional tectonic stresses, which lie in the horizontal plane. Thus, they lie in the x-y plane, rotated by some angle, α, $0 \leq \alpha \leq 90°$. These horizontal stresses are the principal values, therefore normal to each other. It is easily shown, using the principle of virtual work, that as

$$dF_x = S_x d\Gamma \tag{F.185}$$

where Γ is the arbitrary surface boundary, and S_x is the horizontal stress loading. Whence

$$\left.\begin{aligned} F_{x_i} &= \int_{\Gamma^*} N_i S_x \, d\Gamma \\ F_{y_i} &= \int_{\Gamma^*} N_i S_y \, d\Gamma \\ F_{z_i} &= \int_{\Gamma^*} N_i S_z \, d\Gamma \end{aligned}\right\} \tag{F.186}$$

where the vertical loading S_z is optional. If the actual regional principal stresses are rotated in the x-y plane by the angle α, then the x and y stresses are related to these regional stresses by

$$\begin{Bmatrix} S_x \\ S_y \end{Bmatrix} = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} \begin{Bmatrix} S_{H_1} \\ S_{H_2} \end{Bmatrix} \tag{F.187}$$

where S_{H_1} is defined as the regional stress in the x-y plane acting at angle α from the x-axis. The only remaining issue is the implementation of the side integrations. This done using two or three point Gaussian integration, ensuring that the appropriate element side is identified with the correct isoparametric coordinate value.

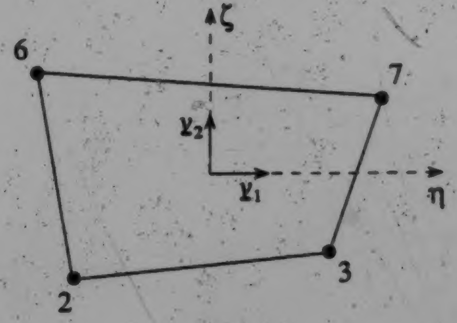


Figure F–2

Local face tangent vectors on side of element.

The area for the integration must be calculated. Consider the arbitrary side being loaded, defined by $\xi = +1$, as shown in Figure F–2. The face tangent vectors, $\vec{v}_1, \vec{v}_2$ are given by (Cook, 1981), in this case,

$$\vec{v}_1 = (J_{21}\hat{\imath} + J_{22}\hat{\jmath} + J_{23}\hat{k})d\eta \tag{F.188a}$$

$$\vec{v}_2 = (J_{31}\hat{\imath} + J_{32}\hat{\jmath} + J_{33}\hat{k})d\zeta \tag{F.188b}$$

where J_{ij} are the elements of the Jacobian of transformation. Now, if $\vec{c} = \vec{a} \times \vec{b}$, then the length (magnitude) of $\vec{c}$ is equal to the area of the parallelogram formed by $\vec{a}$ and $\vec{b}$ (Kreysig, p. 270). Furthermore,

$$\left|\vec{a} \times \vec{b}\right| = \sqrt{(\vec{a}\cdot\vec{a})(\vec{b}\cdot\vec{b}) - (\vec{a}\cdot\vec{b})^2} \tag{F.189}$$

Thus,

$$\begin{aligned} d\Gamma &= \begin{vmatrix} \hat{\imath} & \hat{\jmath} & \hat{k} \\ J_{21} & J_{22} & J_{23} \\ J_{31} & J_{32} & J_{33} \end{vmatrix} d\zeta d\eta \\ &= S\, d\zeta d\eta \end{aligned} \tag{F.190}$$

S can be found using rows k and ℓ from the three-dimensional Jacobian matrix of transformation, where k,ℓ indicate the local coordinates in which the surface lies.

$$S = \{[(\text{row } k)\cdot(\text{row } k)][(\text{row } \ell)\cdot(\text{row } \ell)] - [(\text{row } k)\cdot(\text{row } \ell)]^2\}^2 \tag{F.191}$$

F.4.5 *Accommodating the initial stresses*

The stresses must be calculated, and the resulting nodal loads and reactions at fixed nodes found. When applying a regional stress field the values are calculated as:

$$\begin{Bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \tau_{yz} \\ \tau_{zx} \\ \tau_{xy} \end{Bmatrix} \tag{F.192}$$

If the stresses to be considered are confined to a vertical principal stress and two lying in the horizontal plane, then $\sigma_v \equiv \sigma_{zz}$, and $\tau_{yz} = \tau_{zx} = 0$. Thus, only σ_{H_1} and σ_{H_2} must be resolved into σ_{yy}, σ_{xx}, and τ_{xy}. This resolution can be simply done by specifying the angle between one of the horizontal principal stresses and the x-axis. This corresponds to a rotation, $-\alpha$, to the σ_{xx} (x) axis.

Generalizing to a stress vector on a plane inclined between the x and y axes, resolved into normal and shear stress components on the plane, the rotation formulae are (Nash, 1977):

$$\sigma_n = \frac{\sigma_x + \sigma_y}{2} - \frac{\sigma_x - \sigma_y}{2} \cos 2\alpha + \tau_{xy} \sin 2\alpha \tag{F.193a}$$

$$\tau = \frac{\sigma_x - \sigma_y}{2} \sin 2\alpha + \tau_{xy} \cos 2\alpha \tag{F.193b}$$

As we are rotating from the horizontal principal stresses, in this rotation $\sigma_n \equiv \sigma_{yy}$ and $\tau \equiv \sigma_{xx}$, $\sigma_x \equiv \sigma_{H_1}$, $\sigma_y \equiv \sigma_{H_2}$, and $\tau_{xy} = 0$. The final reactions and displacements satisfying this initial stress regime can be calculated by the finite element program. The vector containing the total displacements can then be set to zero.

The nodal loads F^e corresponding to the initial stress are found from

$$F^e_{\sigma^0} = -\int_{\Omega^e} [\mathbf{B}_i]^T \underline{\sigma}^0 \, d\Omega \tag{F.194}$$

which has the same form as for the thermal loads.

Appendix G

SAMPLE DATA FOR USE WITH PLAST-PT3

This data was used for the staggered run described in Chapter VIII.

```
# This is the comment character
# Debugging flags
# 1 2 3 4 5 6 7 8 9 0
  F F F F F F F F F F
#
# Title (100 chars)
3-D ELASTO-PLASTIC PROBLEM: Mohr-Coulomb, L8 elements FOR Staggered run
# npoin, nelem,  ntype, nmats, ngaus, nalgo, ncrit, nstre
  896    637       4       3       8     2      3       6
# nvfix, nincs(1), nincs(2), incval (kN) (negative for no inc. data), kPa
  428      1         1           5.0e5    2.0e3
#
# Loads applied as a user option:
#    0: stresses applied globally & reaction loads evaluated
#    1: stresses applied as trial values & no reaction loads evaluated
#    2: stresses applied as trial values & reaction loads evaluated
#
# Number of printing flags used, dp, dt as trial stresses?
# npflag stflag(1.2.3)   tstep      rstflag   relax    rlxflag    rlxstrt
   9      0    0    1   -1          -1         1.5       2          5
#
# Output provided on Mon Apr 30 23:16:15 1990
#
#  nelem  nmats  mnode   nfix
#    637     3    896    428
#
# Total number of elements =  637
# Total number of material types =    3
#
# Elem    Material    1     2     3     4     5     6     7     8
     1        3        1     2    10     9    65    66    74    73
     2        3        2     3    11    10    66    67    75    74
     3        3        3     4    12    11    67    68    76    75
     4        3        4     5    13    12    68    69    77    76
     5        3        5     6    14    13    69    70    78    77
     6        3        6     7    15    14    70    71    79    78
     7        3        7     8    16    15    71    72    80    79
     8        3        9    10    18    17    73    74    82    81
     9        3       10    11    19    18    74    75    83    82
    10        3       11    12    20    19    75    76    84    83
    11        3       12    13    21    20    76    77    85    84
    12        3       13    14    22    21    77    78    86    85
    13        3       14    15    23    22    78    79    87    86
    14        3       15    16    24    23    79    80    88    87
    15        3       17    18    26    25    81    82    90    89
    16        3       18    19    27    26    82    83    91    90
    17        3       19    20    28    27    83    84    92    91
    18        3       20    21    29    28    84    85    93    92
    19        3       21    22    30    29    85    86    94    93
    20        3       22    23    31    30    86    87    95    94
    21        3       23    24    32    31    87    88    96    95
```

22	3	25	26	34	33	89	90	98	97
23	3	26	27	35	34	90	91	99	98
24	3	27	28	36	35	91	92	100	99
25	3	28	29	37	36	92	93	101	100
26	3	29	30	38	37	93	94	102	101
27	3	30	31	39	38	94	95	103	102
28	3	31	32	40	39	95	96	104	103
29	3	33	34	42	41	97	98	106	105
30	3	34	35	43	42	98	99	107	106
31	3	35	36	44	43	99	100	108	107
32	3	36	37	45	44	100	101	109	108
33	3	37	38	46	45	101	102	110	109
34	3	38	39	47	46	102	103	111	110
35	3	39	40	48	47	103	104	112	111
36	3	41	42	50	49	105	106	114	113
37	3	42	43	51	50	106	107	115	114
38	3	43	44	52	51	107	108	116	115
39	3	44	45	53	52	108	109	117	116
40	3	45	46	54	53	109	110	118	117
41	3	46	47	55	54	110	111	119	118
42	3	47	48	56	55	111	112	120	119
43	3	49	50	58	57	113	114	122	121
44	3	50	51	59	58	114	115	123	122
45	3	51	52	60	59	115	116	124	123
46	3	52	53	61	60	116	117	125	124
47	3	53	54	62	61	117	118	126	125
48	3	54	55	63	62	118	119	127	126
49	3	55	56	64	63	119	120	128	127
50	3	65	66	74	73	129	130	138	137
51	3	66	67	75	74	130	131	139	138
52	3	67	68	76	75	131	132	140	139
53	3	68	69	77	76	132	133	141	140
54	3	69	70	78	77	133	134	142	141
55	3	70	71	79	78	134	135	143	142
56	3	71	72	80	79	135	136	144	143
57	3	73	74	82	81	137	138	146	145
58	3	74	75	83	82	138	139	147	146
59	3	75	76	84	83	139	140	148	147
60	3	76	77	85	84	140	141	149	148
61	3	77	78	86	85	141	142	150	149
62	3	78	79	87	86	142	143	151	150
63	3	79	80	88	87	143	144	152	151
64	3	81	82	90	89	145	146	154	153
65	3	82	83	91	90	146	147	155	154
66	3	83	84	92	91	147	148	156	155
67	3	84	85	93	92	148	149	157	156
68	3	85	86	94	93	149	150	158	157
69	3	86	87	95	94	150	151	159	158
70	3	87	88	96	95	151	152	160	159
71	3	89	90	98	97	153	154	162	161
72	3	90	91	99	98	154	155	163	162

73	3	91	92	100	99	155	156	164	163
74	3	92	93	101	100	156	157	165	164
75	3	93	94	102	101	157	158	166	165
76	3	94	95	103	102	158	159	167	166
77	3	95	96	104	103	159	160	168	167
78	3	97	98	106	105	161	162	170	169
79	3	98	99	107	106	162	163	171	170
80	3	99	100	108	107	163	164	172	171
81	3	100	101	109	108	164	165	173	172
82	3	101	102	110	109	165	166	174	173
83	3	102	103	111	110	166	167	175	174
84	3	103	104	112	111	167	168	176	175
85	3	105	106	114	113	169	170	178	177
86	3	106	107	115	114	170	171	179	178
87	3	107	108	116	115	171	172	180	179
88	3	108	109	117	116	172	173	181	180
89	3	109	110	118	117	173	174	182	181
90	3	110	111	119	118	174	175	183	182
91	3	111	112	120	119	175	176	184	183
92	3	113	114	122	121	177	178	186	185
93	3	114	115	123	122	178	179	187	186
94	3	115	116	124	123	179	180	188	187
95	3	116	117	125	124	180	181	189	188
96	3	117	118	126	125	181	182	190	189
97	3	118	119	127	126	182	183	191	190
98	3	119	120	128	127	183	184	192	191
99	3	129	130	138	137	193	194	202	201
100	3	130	131	139	138	194	195	203	202
101	3	131	132	140	139	195	196	204	203
102	3	132	133	141	140	196	197	205	204
103	3	133	134	142	141	197	198	206	205
104	3	134	135	143	142	198	199	207	206
105	3	135	136	144	143	199	200	208	207
106	3	137	138	146	145	201	202	210	209
107	3	138	139	147	146	202	203	211	210
108	3	139	140	148	147	203	204	212	211
109	3	140	141	149	148	204	205	213	212
110	3	141	142	150	149	205	206	214	213
111	3	142	143	151	150	206	207	215	214
112	3	143	144	152	151	207	208	216	215
113	3	145	146	154	153	209	210	218	217
114	3	146	147	155	154	210	211	219	218
115	3	147	148	156	155	211	212	220	219
116	3	148	149	157	156	212	213	221	220
117	3	149	150	158	157	213	214	222	221
118	3	150	151	159	158	214	215	223	222
119	3	151	152	160	159	215	216	224	223
120	3	153	154	162	161	217	218	226	225
121	3	154	155	163	162	218	219	227	226
122	3	155	156	164	163	219	220	228	227
123	3	156	157	165	164	220	221	229	228

124	3	157	158	166	165	221	222	230	229
125	3	158	159	167	166	222	223	231	230
126	3	159	160	168	167	223	224	232	231
127	3	161	162	170	169	225	226	234	233
128	3	162	163	171	170	226	227	235	234
129	3	163	164	172	171	227	228	236	235
130	3	164	165	173	172	228	229	237	236
131	3	165	166	174	173	229	230	238	237
132	3	166	167	175	174	230	231	239	238
133	3	167	168	176	175	231	232	240	239
134	3	169	170	178	177	233	234	242	241
135	3	170	171	179	178	234	235	243	242
136	3	171	172	180	179	235	236	244	243
137	3	172	173	181	180	236	237	245	244
138	3	173	174	182	181	237	238	246	245
139	3	174	175	183	182	238	239	247	246
140	3	175	176	184	183	239	240	248	247
141	3	177	178	186	185	241	242	250	249
142	3	178	179	187	186	242	243	251	250
143	3	179	180	188	187	243	244	252	251
144	3	180	181	189	188	244	245	253	252
145	3	181	182	190	189	245	246	254	253
146	3	182	183	191	190	246	247	255	254
147	3	183	184	192	191	247	248	256	255
148	3	193	194	202	201	257	258	266	265
149	3	194	195	203	202	258	259	267	266
150	3	195	196	204	203	259	260	268	267
151	3	196	197	205	204	260	261	269	268
152	3	197	198	206	205	261	262	270	269
153	3	198	199	207	206	262	263	271	270
154	3	199	200	208	207	263	264	272	271
155	3	201	202	210	209	265	266	274	273
156	3	202	203	211	210	266	267	275	274
157	3	203	204	212	211	267	268	276	275
158	3	204	205	213	212	268	269	277	276
159	3	205	206	214	213	269	270	278	277
160	3	206	207	215	214	270	271	279	278
161	3	207	208	216	215	271	272	280	279
162	3	209	210	218	217	273	274	282	281
163	3	210	211	219	218	274	275	283	282
164	3	211	212	220	219	275	276	284	283
165	3	212	213	221	220	276	277	285	284
166	3	213	214	222	221	277	278	286	285
167	3	214	215	223	222	278	279	287	286
168	3	215	216	224	223	279	280	288	287
169	3	217	218	226	225	281	282	290	289
170	3	218	219	227	226	282	283	291	290
171	3	219	220	228	227	283	284	292	291
172	3	220	221	229	228	284	285	293	292
173	3	221	222	230	229	285	286	294	293
174	3	222	223	231	230	286	287	295	294

175	3	223	224	232	231	287	288	296	295
176	3	225	226	234	233	289	290	298	297
177	3	226	227	235	234	290	291	299	298
178	3	227	228	236	235	291	292	300	299
179	3	228	229	237	236	292	293	301	300
180	3	229	230	238	237	293	294	302	301
181	3	230	231	239	238	294	295	303	302
182	3	231	232	240	239	295	296	304	303
183	3	233	234	242	241	297	298	306	305
184	3	234	235	243	242	298	299	307	306
185	3	235	236	244	243	299	300	308	307
186	3	236	237	245	244	300	301	309	308
187	3	237	238	246	245	301	302	310	309
188	3	238	239	247	246	302	303	311	310
189	3	239	240	248	247	303	304	312	311
190	3	241	242	250	249	305	306	314	313
191	3	242	243	251	250	306	307	315	314
192	3	243	244	252	251	307	308	316	315
193	3	244	245	253	252	308	309	317	316
194	3	245	246	254	253	309	310	318	317
195	3	246	247	255	254	310	311	319	318
196	3	247	248	256	255	311	312	320	319
197	1	257	258	266	265	321	322	330	329
198	1	258	259	267	266	322	323	331	330
199	1	259	260	268	267	323	324	332	331
200	1	260	261	269	268	324	325	333	332
201	1	261	262	270	269	325	326	334	333
202	1	262	263	271	270	326	327	335	334
203	1	263	264	272	271	327	328	336	335
204	1	265	266	274	273	329	330	338	337
205	1	266	267	275	274	330	331	339	338
206	1	267	268	276	275	331	332	340	339
207	1	268	269	277	276	332	333	341	340
208	1	269	270	278	277	333	334	342	341
209	1	270	271	279	278	334	335	343	342
210	1	271	272	280	279	335	336	344	343
211	1	273	274	282	281	337	338	346	345
212	1	274	275	283	282	338	339	347	346
213	1	275	276	284	283	339	340	348	347
214	1	276	277	285	284	340	341	349	348
215	1	277	278	286	285	341	342	350	349
216	1	278	279	287	286	342	343	351	350
217	1	279	280	288	287	343	344	352	351
218	1	281	282	290	289	345	346	354	353
219	1	282	283	291	290	346	347	355	354
220	1	283	284	292	291	347	348	356	355
221	1	284	285	293	292	348	349	357	356
222	1	285	286	294	293	349	350	358	357
223	1	286	287	295	294	350	351	359	358
224	1	287	288	296	295	351	352	360	359
225	1	289	290	298	297	353	354	362	361

226	1	290	291	299	298	354	355	363	362
227	1	291	292	300	299	355	356	364	363
228	1	292	293	301	300	356	357	365	364
229	1	293	294	302	301	357	358	366	365
230	1	294	295	303	302	358	359	367	366
231	1	295	296	304	303	359	360	368	367
232	1	297	298	306	305	361	362	370	369
233	1	298	299	307	306	362	363	371	370
234	1	299	300	308	307	363	364	372	371
235	1	300	301	309	308	364	365	373	372
236	1	301	302	310	309	365	366	374	373
237	1	302	303	311	310	366	367	375	374
238	1	303	304	312	311	367	368	376	375
239	1	305	306	314	313	369	370	378	377
240	1	306	307	315	314	370	371	379	378
241	1	307	308	316	315	371	372	380	379
242	1	308	309	317	316	372	373	381	380
243	1	309	310	318	317	373	374	382	381
244	1	310	311	319	318	374	375	383	382
245	1	311	312	320	319	375	376	384	383
246	1	321	322	330	329	385	386	394	393
247	1	322	323	331	330	386	387	395	394
248	1	323	324	332	331	387	388	396	395
249	1	324	325	333	332	388	389	397	396
250	1	325	326	334	333	389	390	398	397
251	1	326	327	335	334	390	391	399	398
252	1	327	328	336	335	391	392	400	399
253	1	329	330	338	337	393	394	402	401
254	1	330	331	339	338	394	395	403	402
255	1	331	332	340	339	395	396	404	403
256	1	332	333	341	340	396	397	405	404
257	1	333	334	342	341	397	398	406	405
258	1	334	335	343	342	398	399	407	406
259	1	335	336	344	343	399	400	408	407
260	1	337	338	346	345	401	402	410	409
261	1	338	339	347	346	402	403	411	410
262	1	339	340	348	347	403	404	412	411
263	1	340	341	349	348	404	405	413	412
264	1	341	342	350	349	405	406	414	413
265	1	342	343	351	350	406	407	415	414
266	1	343	344	352	351	407	408	416	415
267	1	345	346	354	353	409	410	418	417
268	1	346	347	355	354	410	411	419	418
269	1	347	348	356	355	411	412	420	419
270	1	348	349	357	356	412	413	421	420
271	1	349	350	358	357	413	414	422	421
272	1	350	351	359	358	414	415	423	422
273	1	351	352	360	359	415	416	424	423
274	1	353	354	362	361	417	418	426	425
275	1	354	355	363	362	418	419	427	426
276	1	355	356	364	363	419	420	428	427

277	1	356	357	365	364	420	421	429	428
278	1	357	358	366	365	421	422	430	429
279	1	358	359	367	366	422	423	431	430
280	1	359	360	368	367	423	424	432	431
281	1	361	362	370	369	425	426	434	433
282	1	362	363	371	370	426	427	435	434
283	1	363	364	372	371	427	428	436	435
284	1	364	365	373	372	428	429	437	436
285	1	365	366	374	373	429	430	438	437
286	1	366	367	375	374	430	431	439	438
287	1	367	368	376	375	431	432	440	439
288	1	369	370	378	377	433	434	442	441
289	1	370	371	379	378	434	435	443	442
290	1	371	372	380	379	435	436	444	443
291	1	372	373	381	380	436	437	445	444
292	1	373	374	382	381	437	438	446	445
293	1	374	375	383	382	438	439	447	446
294	1	375	376	384	383	439	440	448	447
295	2	385	386	394	393	449	450	458	457
296	2	386	387	395	394	450	451	459	458
297	2	387	388	396	395	451	452	460	459
298	2	388	389	397	396	452	453	461	460
299	2	389	390	398	397	453	454	462	461
300	2	390	391	399	398	454	455	463	462
301	2	391	392	400	399	455	456	464	463
302	2	393	394	402	401	457	458	466	465
303	2	394	395	403	402	458	459	467	466
304	2	395	396	404	403	459	460	468	467
305	2	396	397	405	404	460	461	469	468
306	2	397	398	406	405	461	462	470	469
307	2	398	399	407	406	462	463	471	470
308	2	399	400	408	407	463	464	472	471
309	2	401	402	410	409	465	466	474	473
310	2	402	403	411	410	466	467	475	474
311	2	403	404	412	411	467	468	476	475
312	2	404	405	413	412	468	469	477	476
313	2	405	406	414	413	469	470	478	477
314	2	406	407	415	414	470	471	479	478
315	2	407	408	416	415	471	472	480	479
316	2	409	410	418	417	473	474	482	481
317	2	410	411	419	418	474	475	483	482
318	2	411	412	420	419	475	476	484	483
319	2	412	413	421	420	476	477	485	484
320	2	413	414	422	421	477	478	486	485
321	2	414	415	423	422	478	479	487	486
322	2	415	416	424	423	479	480	488	487
323	2	417	418	426	425	481	482	490	489
324	2	418	419	427	426	482	483	491	490
325	2	419	420	428	427	483	484	492	491
326	2	420	421	429	428	484	485	493	492
327	2	421	422	430	429	485	486	494	493

328	2	422	423	431	430	486	487	495	494
329	2	423	424	432	431	487	488	496	495
330	2	425	426	434	433	489	490	498	497
331	2	426	427	435	434	490	491	499	498
332	2	427	428	436	435	491	492	500	499
333	2	428	429	437	436	492	493	501	500
334	2	429	430	438	437	493	494	502	501
335	2	430	431	439	438	494	495	503	502
336	2	431	432	440	439	495	496	504	503
337	2	433	434	442	441	497	498	506	505
338	2	434	435	443	442	498	499	507	506
339	2	435	436	444	443	499	500	508	507
340	2	436	437	445	444	500	501	509	508
341	2	437	438	446	445	501	502	510	509
342	2	438	439	447	446	502	503	511	510
343	2	439	440	448	447	503	504	512	511
344	2	449	450	458	457	513	514	522	521
345	2	450	451	459	458	514	515	523	522
346	2	451	452	460	459	515	516	524	523
347	2	452	453	461	460	516	517	525	524
348	2	453	454	462	461	517	518	526	525
349	2	454	455	463	462	518	519	527	526
350	2	455	456	464	463	519	520	528	527
351	2	457	458	466	465	521	522	530	529
352	2	458	459	467	466	522	523	531	530
353	2	459	460	468	467	523	524	532	531
354	2	460	461	469	468	524	525	533	532
355	2	461	462	470	469	525	526	534	533
356	2	462	463	471	470	526	527	535	534
357	2	463	464	472	471	527	528	536	535
358	2	465	466	474	473	529	530	538	537
359	2	466	467	475	474	530	531	539	538
360	2	467	468	476	475	531	532	540	539
361	2	468	469	477	476	532	533	541	540
362	2	469	470	478	477	533	534	542	541
363	2	470	471	479	478	534	535	543	542
364	2	471	472	480	479	535	536	544	543
365	2	473	474	482	481	537	538	546	545
366	2	474	475	483	482	538	539	547	546
367	2	475	476	484	483	539	540	548	547
368	2	476	477	485	484	540	541	549	548
369	2	477	478	486	485	541	542	550	549
370	2	478	479	487	486	542	543	551	550
371	2	479	480	488	487	543	544	552	551
372	2	481	482	490	489	545	546	554	553
373	2	482	483	491	490	546	547	555	554
374	2	483	484	492	491	547	548	556	555
375	2	484	485	493	492	548	549	557	556
376	2	485	486	494	493	549	550	558	557
377	2	486	487	495	494	550	551	559	558
378	2	487	488	496	495	551	552	560	559

379	2	489	490	498	497	553	554	562	561
380	2	490	491	499	498	554	555	563	562
381	2	491	492	500	499	555	556	564	563
382	2	492	493	501	500	556	557	565	564
383	2	493	494	502	501	557	558	566	565
384	2	494	495	503	502	558	559	567	566
385	2	495	496	504	503	559	560	568	567
386	2	497	498	506	505	561	562	570	569
387	2	498	499	507	506	562	563	571	570
388	2	499	500	508	507	563	564	572	571
389	2	500	501	509	508	564	565	573	572
390	2	501	502	510	509	565	566	574	573
391	2	502	503	511	510	566	567	575	574
392	2	503	504	512	511	567	568	576	575
393	2	513	514	522	521	577	578	586	585
394	2	514	515	523	522	578	579	587	586
395	2	515	516	524	523	579	580	588	587
396	2	516	517	525	524	580	581	589	588
397	2	517	518	526	525	581	582	590	589
398	2	518	519	527	526	582	583	591	590
399	2	519	520	528	527	583	584	592	591
400	2	521	522	530	529	585	586	594	593
401	2	522	523	531	530	586	587	595	594
402	2	523	524	532	531	587	588	596	595
403	2	524	525	533	532	588	589	597	596
404	2	525	526	534	533	589	590	598	597
405	2	526	527	535	534	590	591	599	598
406	2	527	528	536	535	591	592	600	599
407	2	529	530	538	537	593	594	602	601
408	2	530	531	539	538	594	595	603	602
409	2	531	532	540	539	595	596	604	603
410	2	532	533	541	540	596	597	605	604
411	2	533	534	542	541	597	598	606	605
412	2	534	535	543	542	598	599	607	606
413	2	535	536	544	543	599	600	608	607
414	2	537	538	546	545	601	602	610	609
415	2	538	539	547	546	602	603	611	610
416	2	539	540	548	547	603	604	612	611
417	2	540	541	549	548	604	605	613	612
418	2	541	542	550	549	605	606	614	613
419	2	542	543	551	550	606	607	615	614
420	2	543	544	552	551	607	608	616	615
421	2	545	546	554	553	609	610	618	617
422	2	546	547	555	554	610	611	619	618
423	2	547	548	556	555	611	612	620	619
424	2	548	549	557	556	612	613	621	620
425	2	549	550	558	557	613	614	622	621
426	2	550	551	559	558	614	615	623	622
427	2	551	552	560	559	615	616	624	623
428	2	553	554	562	561	617	618	626	625
429	2	554	555	563	562	618	619	627	626

430	2	555	556	564	563	619	620	628	627
431	2	556	557	565	564	620	621	629	628
432	2	557	558	566	565	621	622	630	629
433	2	558	559	567	566	622	623	631	630
434	2	559	560	568	567	623	624	632	631
435	2	561	562	570	569	625	626	634	633
436	2	562	563	571	570	626	627	635	634
437	2	563	564	572	571	627	628	636	635
438	2	564	565	573	572	628	629	637	636
439	2	565	566	574	573	629	630	638	637
440	2	566	567	575	574	630	631	639	638
441	2	567	568	576	575	631	632	640	639
442	2	577	578	586	585	641	642	650	649
443	2	578	579	587	586	642	643	651	650
444	2	579	580	588	587	643	644	652	651
445	2	580	581	589	588	644	645	653	652
446	2	581	582	590	589	645	646	654	653
447	2	582	583	591	590	646	647	655	654
448	2	583	584	592	591	647	648	656	655
449	2	585	586	594	593	649	650	658	657
450	2	586	587	595	594	650	651	659	658
451	2	587	588	596	595	651	652	660	659
452	2	588	589	597	596	652	653	661	660
453	2	589	590	598	597	653	654	662	661
454	2	590	591	599	598	654	655	663	662
455	2	591	592	600	599	655	656	664	663
456	2	593	594	602	601	657	658	666	665
457	2	594	595	603	602	658	659	667	666
458	2	595	596	604	603	659	660	668	667
459	2	596	597	605	604	660	661	669	668
460	2	597	598	606	605	661	662	670	669
461	2	598	599	607	606	662	663	671	670
462	2	599	600	608	607	663	664	672	671
463	2	601	602	610	609	665	666	674	673
464	2	602	603	611	610	666	667	675	674
465	2	603	604	612	611	667	668	676	675
466	2	604	605	613	612	668	669	677	676
467	2	605	606	614	613	669	670	678	677
468	2	606	607	615	614	670	671	679	678
469	2	607	608	616	615	671	672	680	679
470	2	609	610	618	617	673	674	682	681
471	2	610	611	619	618	674	675	683	682
472	2	611	612	620	619	675	676	684	683
473	2	612	613	621	620	676	677	685	684
474	2	613	614	622	621	677	678	686	685
475	2	614	615	623	622	678	679	687	686
476	2	615	616	624	623	679	680	688	687
477	2	617	618	626	625	681	682	690	689
478	2	618	619	627	626	682	683	691	690
479	2	619	620	628	627	683	684	692	691
480	2	620	621	629	628	684	685	693	692

481	2	621	622	630	629	685	686	694	693
482	2	622	623	631	630	686	687	695	694
483	2	623	624	632	631	687	688	696	695
484	2	625	626	634	633	689	690	698	697
485	2	626	627	635	634	690	691	699	698
486	2	627	628	636	635	691	692	700	699
487	2	628	629	637	636	692	693	701	700
488	2	629	630	638	637	693	694	702	701
489	2	630	631	639	638	694	695	703	702
490	2	631	632	640	639	695	696	704	703
491	2	641	642	650	649	705	706	714	713
492	2	642	643	651	650	706	707	715	714
493	2	643	644	652	651	707	708	716	715
494	2	644	645	653	652	708	709	717	716
495	2	645	646	654	653	709	710	718	717
496	2	646	647	655	654	710	711	719	718
497	2	647	648	656	655	711	712	720	719
498	2	649	650	658	657	713	714	722	721
499	2	650	651	659	658	714	715	723	722
500	2	651	652	660	659	715	716	724	723
501	2	652	653	661	660	716	717	725	724
502	2	653	654	662	661	717	718	726	725
503	2	654	655	663	662	718	719	727	726
504	2	655	656	664	663	719	720	728	727
505	2	657	658	666	665	721	722	730	729
506	2	658	659	667	666	722	723	731	730
507	2	659	660	668	667	723	724	732	731
508	2	660	661	669	668	724	725	733	732
509	2	661	662	670	669	725	726	734	733
510	2	662	663	671	670	726	727	735	734
511	2	663	664	672	671	727	728	736	735
512	2	665	666	674	673	729	730	738	737
513	2	666	667	675	674	730	731	739	738
514	2	667	668	676	675	731	732	740	739
515	2	668	669	677	676	732	733	741	740
516	2	669	670	678	677	733	734	742	741
517	2	670	671	679	678	734	735	743	742
518	2	671	672	680	679	735	736	744	743
519	2	673	674	682	681	737	738	746	745
520	2	674	675	683	682	738	739	747	746
521	2	675	676	684	683	739	740	748	747
522	2	676	677	685	684	740	741	749	748
523	2	677	678	686	685	741	742	750	749
524	2	678	679	687	686	742	743	751	750
525	2	679	680	688	687	743	744	752	751
526	2	681	682	690	689	745	746	754	753
527	2	682	683	691	690	746	747	755	754
528	2	683	684	692	691	747	748	756	755
529	2	684	685	693	692	748	749	757	756
530	2	685	686	694	693	749	750	758	757
531	2	686	687	695	694	750	751	759	758

532	2	687	688	696	695	751	752	760	759
533	2	689	690	698	697	753	754	762	761
534	2	690	691	699	698	754	755	763	762
535	2	691	692	700	699	755	756	764	763
536	2	692	693	701	700	756	757	765	764
537	2	693	694	702	701	757	758	766	765
538	2	694	695	703	702	758	759	767	766
539	2	695	696	704	703	759	760	768	767
540	2	705	706	714	713	769	770	778	777
541	2	706	707	715	714	770	771	779	778
542	2	707	708	716	715	771	772	780	779
543	2	708	709	717	716	772	773	781	780
544	2	709	710	718	717	773	774	782	781
545	2	710	711	719	718	774	775	783	782
546	2	711	712	720	719	775	776	784	783
547	2	713	714	722	721	777	778	786	785
548	2	714	715	723	722	778	779	787	786
549	2	715	716	724	723	779	780	788	787
550	2	716	717	725	724	780	781	789	788
551	2	717	718	726	725	781	782	790	789
552	2	718	719	727	726	782	783	791	790
553	2	719	720	728	727	783	784	792	791
554	2	721	722	730	729	785	786	794	793
555	2	722	723	731	730	786	787	795	794
556	2	723	724	732	731	787	788	796	795
557	2	724	725	733	732	788	789	797	796
558	2	725	726	734	733	789	790	798	797
559	2	726	727	735	734	790	791	799	798
560	2	727	728	736	735	791	792	800	799
561	2	729	730	738	737	793	794	802	801
562	2	730	731	739	738	794	795	803	802
563	2	731	732	740	739	795	796	804	803
564	2	732	733	741	740	796	797	805	804
565	2	733	734	742	741	797	798	806	805
566	2	734	735	743	742	798	799	807	806
567	2	735	736	744	743	799	800	808	807
568	2	737	738	746	745	801	802	810	809
569	2	738	739	747	746	802	803	811	810
570	2	739	740	748	747	803	804	812	811
571	2	740	741	749	748	804	805	813	812
572	2	741	742	750	749	805	806	814	813
573	2	742	743	751	750	806	807	815	814
574	2	743	744	752	751	807	808	816	815
575	2	745	746	754	753	809	810	818	817
576	2	746	747	755	754	810	811	819	818
577	2	747	748	756	755	811	812	820	819
578	2	748	749	757	756	812	813	821	820
579	2	749	750	758	757	813	814	822	821
580	2	750	751	759	758	814	815	823	822
581	2	751	752	760	759	815	816	824	823
582	2	753	754	762	761	817	818	826	825

583	2	754	755	763	762	818	819	827	826
584	2	755	756	764	763	819	820	828	827
585	2	756	757	765	764	820	821	829	828
586	2	757	758	766	765	821	822	830	829
587	2	758	759	767	766	822	823	831	830
588	2	759	760	768	767	823	824	832	831
589	2	769	770	778	777	833	834	842	841
590	2	770	771	779	778	834	835	843	842
591	2	771	772	780	779	835	836	844	843
592	2	772	773	781	780	836	837	845	844
593	2	773	774	782	781	837	838	846	845
594	2	774	775	783	782	838	839	847	846
595	2	775	776	784	783	839	840	848	847
596	2	777	778	786	785	841	842	850	849
597	2	778	779	787	786	842	843	851	850
598	2	779	780	788	787	843	844	852	851
599	2	780	781	789	788	844	845	853	852
600	2	781	782	790	789	845	846	854	853
601	2	782	783	791	790	846	847	855	854
602	2	783	784	792	791	847	848	856	855
603	2	785	786	794	793	849	850	858	857
604	2	786	787	795	794	850	851	859	858
605	2	787	788	796	795	851	852	860	859
606	2	788	789	797	796	852	853	861	860
607	2	789	790	798	797	853	854	862	861
608	2	790	791	799	798	854	855	863	862
609	2	791	792	800	799	855	856	864	863
610	2	793	794	802	801	857	858	866	865
611	2	794	795	803	802	858	859	867	866
612	2	795	796	804	803	859	860	868	867
613	2	796	797	805	804	860	861	869	868
614	2	797	798	806	805	861	862	870	869
615	2	798	799	807	806	862	863	871	870
616	2	799	800	808	807	863	864	872	871
617	2	801	802	810	809	865	866	874	873
618	2	802	803	811	810	866	867	875	874
619	2	803	804	812	811	867	868	876	875
620	2	804	805	813	812	868	869	877	876
621	2	805	806	814	813	869	870	878	877
622	2	806	807	815	814	870	871	879	878
623	2	807	808	816	815	871	872	880	879
624	2	809	810	818	817	873	874	882	881
625	2	810	811	819	818	874	875	883	882
626	2	811	812	820	819	875	876	884	883
627	2	812	813	821	820	876	877	885	884
628	2	813	814	822	821	877	878	886	885
629	2	814	815	823	822	878	879	887	886
630	2	815	816	824	823	879	880	888	887
631	2	817	818	826	825	881	882	890	889
632	2	818	819	827	826	882	883	891	890
633	2	819	820	828	827	883	884	892	891

```
   634      2      820   821   829   828   884   885   893   892
   635      2      821   822   830   829   885   886   894   893
   636      2      822   823   831   830   886   887   895   894
   637      2      823   824   832   831   887   888   896   895
#
# Total number of nodes =  896
#
# Nodal coordinates
# Node      x         y       z
     1     0.0      0.0  -535.4
     2    40.0      0.0  -535.4
     3    80.0      0.0  -535.4
     4    84.0      0.0  -535.4
     5    94.0      0.0  -535.4
     6   114.0      0.0  -535.4
     7   154.0      0.0  -535.4
     8   254.0      0.0  -535.4
     9     0.0     40.0  -535.4
    10    40.0     40.0  -535.4
    11    80.0     40.0  -535.4
    12    84.0     40.0  -535.4
    13    94.0     40.0  -535.4
    14   114.0     40.0  -535.4
    15   154.0     40.0  -535.4
    16   254.0     40.0  -535.4
    17     0.0     80.0  -535.4
    18    40.0     80.0  -535.4
    19    80.0     80.0  -535.4
    20    84.0     80.0  -535.4
    21    94.0     80.0  -535.4
    22   114.0     80.0  -535.4
    23   154.0     80.0  -535.4
    24   254.0     80.0  -535.4
    25     0.0     84.0  -535.4
    26    40.0     84.0  -535.4
    27    80.0     84.0  -535.4
    28    84.0     84.0  -535.4
    29    94.0     84.0  -535.4
    30   114.0     84.0  -535.4
    31   154.0     84.0  -535.4
    32   254.0     84.0  -535.4
    33     0.0     94.0  -535.4
    34    40.0     94.0  -535.4
    35    80.0     94.0  -535.4
    36    84.0     94.0  -535.4
    37    94.0     94.0  -535.4
    38   114.0     94.0  -535.4
    39   154.0     94.0  -535.4
    40   254.0     94.0  -535.4
    41     0.0    114.0  -535.4
    42    40.0    114.0  -535.4
```

43	80.0	114.0	-535.4
44	84.0	114.0	-535.4
45	94.0	114.0	-535.4
46	114.0	114.0	-535.4
47	154.0	114.0	-535.4
48	254.0	114.0	-535.4
49	0.0	154.0	-535.4
50	40.0	154.0	-535.4
51	80.0	154.0	-535.4
52	84.0	154.0	-535.4
53	94.0	154.0	-535.4
54	114.0	154.0	-535.4
55	154.0	154.0	-535.4
56	254.0	154.0	-535.4
57	0.0	254.0	-535.4
58	40.0	254.0	-535.4
59	80.0	254.0	-535.4
60	84.0	254.0	-535.4
61	94.0	254.0	-535.4
62	114.0	254.0	-535.4
63	154.0	254.0	-535.4
64	254.0	254.0	-535.4
65	0.0	0.0	-495.4
66	40.0	0.0	-495.4
67	80.0	0.0	-495.4
68	84.0	0.0	-495.4
69	94.0	0.0	-495.4
70	114.0	0.0	-495.4
71	154.0	0.0	-495.4
72	254.0	0.0	-495.4
73	0.0	40.0	-495.4
74	40.0	40.0	-495.4
75	80.0	40.0	-495.4
76	84.0	40.0	-495.4
77	94.0	40.0	-495.4
78	114.0	40.0	-495.4
79	154.0	40.0	-495.4
80	254.0	40.0	-495.4
81	0.0	80.0	-495.4
82	40.0	80.0	-495.4
83	80.0	80.0	-495.4
84	84.0	80.0	-495.4
85	94.0	80.0	-495.4
86	114.0	80.0	-495.4
87	154.0	80.0	-495.4
88	254.0	80.0	-495.4
89	0.0	84.0	-495.4
90	40.0	84.0	-495.4
91	80.0	84.0	-495.4
92	84.0	84.0	-495.4
93	94.0	84.0	-495.4

94	114.0	84.0	-495.4
95	154.0	84.0	-495.4
96	254.0	84.0	-495.4
97	0.0	94.0	-495.4
98	40.0	94.0	-495.4
99	80.0	94.0	-495.4
100	84.0	94.0	-495.4
101	94.0	94.0	-495.4
102	114.0	94.0	-495.4
103	154.0	94.0	-495.4
104	254.0	94.0	-495.4
105	0.0	114.0	-495.4
106	40.0	114.0	-495.4
107	80.0	114.0	-495.4
108	84.0	114.0	-495.4
109	94.0	114.0	-495.4
110	114.0	114.0	-495.4
111	154.0	114.0	-495.4
112	254.0	114.0	-495.4
113	0.0	154.0	-495.4
114	40.0	154.0	-495.4
115	80.0	154.0	-495.4
116	84.0	154.0	-495.4
117	94.0	154.0	-495.4
118	114.0	154.0	-495.4
119	154.0	154.0	-495.4
120	254.0	154.0	-495.4
121	0.0	254.0	-495.4
122	40.0	254.0	-495.4
123	80.0	254.0	-495.4
124	84.0	254.0	-495.4
125	94.0	254.0	-495.4
126	114.0	254.0	-495.4
127	154.0	254.0	-495.4
128	254.0	254.0	-495.4
129	0.0	0.0	-475.4
130	40.0	0.0	-475.4
131	80.0	0.0	-475.4
132	84.0	0.0	-475.4
133	94.0	0.0	-475.4
134	114.0	0.0	-475.4
135	154.0	0.0	-475.4
136	254.0	0.0	-475.4
137	0.0	40.0	-475.4
138	40.0	40.0	-475.4
139	80.0	40.0	-475.4
140	84.0	40.0	-475.4
141	94.0	40.0	-475.4
142	114.0	40.0	-475.4
143	154.0	40.0	-475.4
144	254.0	40.0	-475.4

145	0.0	80.0	-475.4
146	40.0	80.0	-475.4
147	80.0	80.0	-475.4
148	84.0	80.0	-475.4
149	94.0	80.0	-475.4
150	114.0	80.0	-475.4
151	154.0	80.0	-475.4
152	254.0	80.0	-475.4
153	0.0	84.0	-475.4
154	40.0	84.0	-475.4
155	80.0	84.0	-475.4
156	84.0	84.0	-475.4
157	94.0	84.0	-475.4
158	114.0	84.0	-475.4
159	154.0	84.0	-475.4
160	254.0	84.0	-475.4
161	0.0	94.0	-475.4
162	40.0	94.0	-475.4
163	80.0	94.0	-475.4
164	84.0	94.0	-475.4
165	94.0	94.0	-475.4
166	114.0	94.0	-475.4
167	154.0	94.0	-475.4
168	254.0	94.0	-475.4
169	0.0	114.0	-475.4
170	40.0	114.0	-475.4
171	80.0	114.0	-475.4
172	84.0	114.0	-475.4
173	94.0	114.0	-475.4
174	114.0	114.0	-475.4
175	154.0	114.0	-475.4
176	254.0	114.0	-475.4
177	0.0	154.0	-475.4
178	40.0	154.0	-475.4
179	80.0	154.0	-475.4
180	84.0	154.0	-475.4
181	94.0	154.0	-475.4
182	114.0	154.0	-475.4
183	154.0	154.0	-475.4
184	254.0	154.0	-475.4
185	0.0	254.0	-475.4
186	40.0	254.0	-475.4
187	80.0	254.0	-475.4
188	84.0	254.0	-475.4
189	94.0	254.0	-475.4
190	114.0	254.0	-475.4
191	154.0	254.0	-475.4
192	254.0	254.0	-475.4
193	0.0	0.0	-465.4
194	40.0	0.0	-465.4
195	80.0	0.0	-465.4

196	84.0	0.0	-465.4
197	94.0	0.0	-465.4
198	114.0	0.0	-465.4
199	154.0	0.0	-465.4
200	254.0	0.0	-465.4
201	0.0	40.0	-465.4
202	40.0	40.0	-465.4
203	80.0	40.0	-465.4
204	84.0	40.0	-465.4
205	94.0	40.0	-465.4
206	114.0	40.0	-465.4
207	154.0	40.0	-465.4
208	254.0	40.0	-465.4
209	0.0	80.0	-465.4
210	40.0	80.0	-465.4
211	80.0	80.0	-465.4
212	84.0	80.0	-465.4
213	94.0	80.0	-465.4
214	114.0	80.0	-465.4
215	154.0	80.0	-465.4
216	254.0	80.0	-465.4
217	0.0	84.0	-465.4
218	40.0	84.0	-465.4
219	80.0	84.0	-465.4
220	84.0	84.0	-465.4
221	94.0	84.0	-465.4
222	114.0	84.0	-465.4
223	154.0	84.0	-465.4
224	254.0	84.0	-465.4
225	0.0	94.0	-465.4
226	40.0	94.0	-465.4
227	80.0	94.0	-465.4
228	84.0	94.0	-465.4
229	94.0	94.0	-465.4
230	114.0	94.0	-465.4
231	154.0	94.0	-465.4
232	254.0	94.0	-465.4
233	0.0	114.0	-465.4
234	40.0	114.0	-465.4
235	80.0	114.0	-465.4
236	84.0	114.0	-465.4
237	94.0	114.0	-465.4
238	114.0	114.0	-465.4
239	154.0	114.0	-465.4
240	254.0	114.0	-465.4
241	0.0	154.0	-465.4
242	40.0	154.0	-465.4
243	80.0	154.0	-465.4
244	84.0	154.0	-465.4
245	94.0	154.0	-465.4
246	114.0	154.0	-465.4

247	154.0	154.0	-465.4
248	254.0	154.0	-465.4
249	0.0	254.0	-465.4
250	40.0	254.0	-465.4
251	80.0	254.0	-465.4
252	84.0	254.0	-465.4
253	94.0	254.0	-465.4
254	114.0	254.0	-465.4
255	154.0	254.0	-465.4
256	254.0	254.0	-465.4
257	0.0	0.0	-461.4
258	40.0	0.0	-461.4
259	80.0	0.0	-461.4
260	84.0	0.0	-461.4
261	94.0	0.0	-461.4
262	114.0	0.0	-461.4
263	154.0	0.0	-461.4
264	254.0	0.0	-461.4
265	0.0	40.0	-461.4
266	40.0	40.0	-461.4
267	80.0	40.0	-461.4
268	84.0	40.0	-461.4
269	94.0	40.0	-461.4
270	114.0	40.0	-461.4
271	154.0	40.0	-461.4
272	254.0	40.0	-461.4
273	0.0	80.0	-461.4
274	40.0	80.0	-461.4
275	80.0	80.0	-461.4
276	84.0	80.0	-461.4
277	94.0	80.0	-461.4
278	114.0	80.0	-461.4
279	154.0	80.0	-461.4
280	254.0	80.0	-461.4
281	0.0	84.0	-461.4
282	40.0	84.0	-461.4
283	80.0	84.0	-461.4
284	84.0	84.0	-461.4
285	94.0	84.0	-461.4
286	114.0	84.0	-461.4
287	154.0	84.0	-461.4
288	254.0	84.0	-461.4
289	0.0	94.0	-461.4
290	40.0	94.0	-461.4
291	80.0	94.0	-461.4
292	84.0	94.0	-461.4
293	94.0	94.0	-461.4
294	114.0	94.0	-461.4
295	154.0	94.0	-461.4
296	254.0	94.0	-461.4
297	0.0	114.0	-461.4

298	40.0	114.0	-461.4
299	80.0	114.0	-461.4
300	84.0	114.0	-461.4
301	94.0	114.0	-461.4
302	114.0	114.0	-461.4
303	154.0	114.0	-461.4
304	254.0	114.0	-461.4
305	0.0	154.0	-461.4
306	40.0	154.0	-461.4
307	80.0	154.0	-461.4
308	84.0	154.0	-461.4
309	94.0	154.0	-461.4
310	114.0	154.0	-461.4
311	154.0	154.0	-461.4
312	254.0	154.0	-461.4
313	0.0	254.0	-461.4
314	40.0	254.0	-461.4
315	80.0	254.0	-461.4
316	84.0	254.0	-461.4
317	94.0	254.0	-461.4
318	114.0	254.0	-461.4
319	154.0	254.0	-461.4
320	254.0	254.0	-461.4
321	0.0	0.0	-457.7
322	40.0	0.0	-457.7
323	80.0	0.0	-457.7
324	84.0	0.0	-457.7
325	94.0	0.0	-457.7
326	114.0	0.0	-457.7
327	154.0	0.0	-457.7
328	254.0	0.0	-457.7
329	0.0	40.0	-457.7
330	40.0	40.0	-457.7
331	80.0	40.0	-457.7
332	84.0	40.0	-457.7
333	94.0	40.0	-457.7
334	114.0	40.0	-457.7
335	154.0	40.0	-457.7
336	254.0	40.0	-457.7
337	0.0	80.0	-457.7
338	40.0	80.0	-457.7
339	80.0	80.0	-457.7
340	84.0	80.0	-457.7
341	94.0	80.0	-457.7
342	114.0	80.0	-457.7
343	154.0	80.0	-457.7
344	254.0	80.0	-457.7
345	0.0	84.0	-457.7
346	40.0	84.0	-457.7
347	80.0	84.0	-457.7
348	84.0	84.0	-457.7

349	94.0	84.0	-457.7
350	114.0	84.0	-457.7
351	154.0	84.0	-457.7
352	254.0	84.0	-457.7
353	0.0	94.0	-457.7
354	40.0	94.0	-457.7
355	80.0	94.0	-457.7
356	84.0	94.0	-457.7
357	94.0	94.0	-457.7
358	114.0	94.0	-457.7
359	154.0	94.0	-457.7
360	254.0	94.0	-457.7
361	0.0	114.0	-457.7
362	40.0	114.0	-457.7
363	80.0	114.0	-457.7
364	84.0	114.0	-457.7
365	94.0	114.0	-457.7
366	114.0	114.0	-457.7
367	154.0	114.0	-457.7
368	254.0	114.0	-457.7
369	0.0	154.0	-457.7
370	40.0	154.0	-457.7
371	80.0	154.0	-457.7
372	84.0	154.0	-457.7
373	94.0	154.0	-457.7
374	114.0	154.0	-457.7
375	154.0	154.0	-457.7
376	254.0	154.0	-457.7
377	0.0	254.0	-457.7
378	40.0	254.0	-457.7
379	80.0	254.0	-457.7
380	84.0	254.0	-457.7
381	94.0	254.0	-457.7
382	114.0	254.0	-457.7
383	154.0	254.0	-457.7
384	254.0	254.0	-457.7
385	0.0	0.0	-454.0
386	40.0	0.0	-454.0
387	80.0	0.0	-454.0
388	84.0	0.0	-454.0
389	94.0	0.0	-454.0
390	114.0	0.0	-454.0
391	154.0	0.0	-454.0
392	254.0	0.0	-454.0
393	0.0	40.0	-454.0
394	40.0	40.0	-454.0
395	80.0	40.0	-454.0
396	84.0	40.0	-454.0
397	94.0	40.0	-454.0
398	114.0	40.0	-454.0
399	154.0	40.0	-454.0

400	254.0	40.0	-454.0
401	0.0	80.0	-454.0
402	40.0	80.0	-454.0
403	80.0	80.0	-454.0
404	84.0	80.0	-454.0
405	94.0	80.0	-454.0
406	114.0	80.0	-454.0
407	154.0	80.0	-454.0
408	254.0	80.0	-454.0
409	0.0	84.0	-454.0
410	40.0	84.0	-454.0
411	80.0	84.0	-454.0
412	84.0	84.0	-454.0
413	94.0	84.0	-454.0
414	114.0	84.0	-454.0
415	154.0	84.0	-454.0
416	254.0	84.0	-454.0
417	0.0	94.0	-454.0
418	40.0	94.0	-454.0
419	80.0	94.0	-454.0
420	84.0	94.0	-454.0
421	94.0	94.0	-454.0
422	114.0	94.0	-454.0
423	154.0	94.0	-454.0
424	254.0	94.0	-454.0
425	0.0	114.0	-454.0
426	40.0	114.0	-454.0
427	80.0	114.0	-454.0
428	84.0	114.0	-454.0
429	94.0	114.0	-454.0
430	114.0	114.0	-454.0
431	154.0	114.0	-454.0
432	254.0	114.0	-454.0
433	0.0	154.0	-454.0
434	40.0	154.0	-454.0
435	80.0	154.0	-454.0
436	84.0	154.0	-454.0
437	94.0	154.0	-454.0
438	114.0	154.0	-454.0
439	154.0	154.0	-454.0
440	254.0	154.0	-454.0
441	0.0	254.0	-454.0
442	40.0	254.0	-454.0
443	80.0	254.0	-454.0
444	84.0	254.0	-454.0
445	94.0	254.0	-454.0
446	114.0	254.0	-454.0
447	154.0	254.0	-454.0
448	254.0	254.0	-454.0
449	0.0	0.0	-450.0
450	40.0	0.0	-450.0

451	80.0	0.0	-450.0
452	84.0	0.0	-450.0
453	94.0	0.0	-450.0
454	114.0	0.0	-450.0
455	154.0	0.0	-450.0
456	254.0	0.0	-450.0
457	0.0	40.0	-450.0
458	40.0	40.0	-450.0
459	80.0	40.0	-450.0
460	84.0	40.0	-450.0
461	94.0	40.0	-450.0
462	114.0	40.0	-450.0
463	154.0	40.0	-450.0
464	254.0	40.0	-450.0
465	0.0	80.0	-450.0
466	40.0	80.0	-450.0
467	80.0	80.0	-450.0
468	84.0	80.0	-450.0
469	94.0	80.0	-450.0
470	114.0	80.0	-450.0
471	154.0	80.0	-450.0
472	254.0	80.0	-450.0
473	0.0	84.0	-450.0
474	40.0	84.0	-450.0
475	80.0	84.0	-450.0
476	84.0	84.0	-450.0
477	94.0	84.0	-450.0
478	114.0	84.0	-450.0
479	154.0	84.0	-450.0
480	254.0	84.0	-450.0
481	0.0	94.0	-450.0
482	40.0	94.0	-450.0
483	80.0	94.0	-450.0
484	84.0	94.0	-450.0
485	94.0	94.0	-450.0
486	114.0	94.0	-450.0
487	154.0	94.0	-450.0
488	254.0	94.0	-450.0
489	0.0	114.0	-450.0
490	40.0	114.0	-450.0
491	80.0	114.0	-450.0
492	84.0	114.0	-450.0
493	94.0	114.0	-450.0
494	114.0	114.0	-450.0
495	154.0	114.0	-450.0
496	254.0	114.0	-450.0
497	0.0	154.0	-450.0
498	40.0	154.0	-450.0
499	80.0	154.0	-450.0
500	84.0	154.0	-450.0
501	94.0	154.0	-450.0

502	114.0	154.0	-450.0
503	154.0	154.0	-450.0
504	254.0	154.0	-450.0
505	0.0	254.0	-450.0
506	40.0	254.0	-450.0
507	80.0	254.0	-450.0
508	84.0	254.0	-450.0
509	94.0	254.0	-450.0
510	114.0	254.0	-450.0
511	154.0	254.0	-450.0
512	254.0	254.0	-450.0
513	0.0	0.0	-440.0
514	40.0	0.0	-440.0
515	80.0	0.0	-440.0
516	84.0	0.0	-440.0
517	94.0	0.0	-440.0
518	114.0	0.0	-440.0
519	154.0	0.0	-440.0
520	254.0	0.0	-440.0
521	0.0	40.0	-440.0
522	40.0	40.0	-440.0
523	80.0	40.0	-440.0
524	84.0	40.0	-440.0
525	94.0	40.0	-440.0
526	114.0	40.0	-440.0
527	154.0	40.0	-440.0
528	254.0	40.0	-440.0
529	0.0	80.0	-440.0
530	40.0	80.0	-440.0
531	80.0	80.0	-440.0
532	84.0	80.0	-440.0
533	94.0	80.0	-440.0
534	114.0	80.0	-440.0
535	154.0	80.0	-440.0
536	254.0	80.0	-440.0
537	0.0	84.0	-440.0
538	40.0	84.0	-440.0
539	80.0	84.0	-440.0
540	84.0	84.0	-440.0
541	94.0	84.0	-440.0
542	114.0	84.0	-440.0
543	154.0	84.0	-440.0
544	254.0	84.0	-440.0
545	0.0	94.0	-440.0
546	40.0	94.0	-440.0
547	80.0	94.0	-440.0
548	84.0	94.0	-440.0
549	94.0	94.0	-440.0
550	114.0	94.0	-440.0
551	154.0	94.0	-440.0
552	254.0	94.0	-440.0

553	0.0	114.0	-440.0
554	40.0	114.0	-440.0
555	80.0	114.0	-440.0
556	84.0	114.0	-440.0
557	94.0	114.0	-440.0
558	114.0	114.0	-440.0
559	154.0	114.0	-440.0
560	254.0	114.0	-440.0
561	0.0	154.0	-440.0
562	40.0	154.0	-440.0
563	80.0	154.0	-440.0
564	84.0	154.0	-440.0
565	94.0	154.0	-440.0
566	114.0	154.0	-440.0
567	154.0	154.0	-440.0
568	254.0	154.0	-440.0
569	0.0	254.0	-440.0
570	40.0	254.0	-440.0
571	80.0	254.0	-440.0
572	84.0	254.0	-440.0
573	94.0	254.0	-440.0
574	114.0	254.0	-440.0
575	154.0	254.0	-440.0
576	254.0	254.0	-440.0
577	0.0	0.0	-420.0
578	40.0	0.0	-420.0
579	80.0	0.0	-420.0
580	84.0	0.0	-420.0
581	94.0	0.0	-420.0
582	114.0	0.0	-420.0
583	154.0	0.0	-420.0
584	254.0	0.0	-420.0
585	0.0	40.0	-420.0
586	40.0	40.0	-420.0
587	80.0	40.0	-420.0
588	84.0	40.0	-420.0
589	94.0	40.0	-420.0
590	114.0	40.0	-420.0
591	154.0	40.0	-420.0
592	254.0	40.0	-420.0
593	0.0	80.0	-420.0
594	40.0	80.0	-420.0
595	80.0	80.0	-420.0
596	84.0	80.0	-420.0
597	94.0	80.0	-420.0
598	114.0	80.0	-420.0
599	154.0	80.0	-420.0
600	254.0	80.0	-420.0
601	0.0	84.0	-420.0
602	40.0	84.0	-420.0
603	80.0	84.0	-420.0

604	84.0	84.0	-420.0
605	94.0	84.0	-420.0
606	114.0	84.0	-420.0
607	154.0	84.0	-420.0
608	254.0	84.0	-420.0
609	0.0	94.0	-420.0
610	40.0	94.0	-420.0
611	80.0	94.0	-420.0
612	84.0	94.0	-420.0
613	94.0	94.0	-420.0
614	114.0	94.0	-420.0
615	154.0	94.0	-420.0
616	254.0	94.0	-420.0
617	0.0	114.0	-420.0
618	40.0	114.0	-420.0
619	80.0	114.0	-420.0
620	84.0	114.0	-420.0
621	94.0	114.0	-420.0
622	114.0	114.0	-420.0
623	154.0	114.0	-420.0
624	254.0	114.0	-420.0
625	0.0	154.0	-420.0
626	40.0	154.0	-420.0
627	80.0	154.0	-420.0
628	84.0	154.0	-420.0
629	94.0	154.0	-420.0
630	114.0	154.0	-420.0
631	154.0	154.0	-420.0
632	254.0	154.0	-420.0
633	0.0	254.0	-420.0
634	40.0	254.0	-420.0
635	80.0	254.0	-420.0
636	84.0	254.0	-420.0
637	94.0	254.0	-420.0
638	114.0	254.0	-420.0
639	154.0	254.0	-420.0
640	254.0	254.0	-420.0
641	0.0	0.0	-380.0
642	40.0	0.0	-380.0
643	80.0	0.0	-380.0
644	84.0	0.0	-380.0
645	94.0	0.0	-380.0
646	114.0	0.0	-380.0
647	154.0	0.0	-380.0
648	254.0	0.0	-380.0
649	0.0	40.0	-380.0
650	40.0	40.0	-380.0
651	80.0	40.0	-380.0
652	84.0	40.0	-380.0
653	94.0	40.0	-380.0
654	114.0	40.0	-380.0

655	154.0	40.0	-380.0
656	254.0	40.0	-380.0
657	0.0	80.0	-380.0
658	40.0	80.0	-380.0
659	80.0	80.0	-380.0
660	84.0	80.0	-380.0
661	94.0	80.0	-380.0
662	114.0	80.0	-380.0
663	154.0	80.0	-380.0
664	254.0	80.0	-380.0
665	0.0	84.0	-380.0
666	40.0	84.0	-380.0
667	80.0	84.0	-380.0
668	84.0	84.0	-380.0
669	94.0	84.0	-380.0
670	114.0	84.0	-380.0
671	154.0	84.0	-380.0
672	254.0	84.0	-380.0
673	0.0	94.0	-380.0
674	40.0	94.0	-380.0
675	80.0	94.0	-380.0
676	84.0	94.0	-380.0
677	94.0	94.0	-380.0
678	114.0	94.0	-380.0
679	154.0	94.0	-380.0
680	254.0	94.0	-380.0
681	0.0	114.0	-380.0
682	40.0	114.0	-380.0
683	80.0	114.0	-380.0
684	84.0	114.0	-380.0
685	94.0	114.0	-380.0
686	114.0	114.0	-380.0
687	154.0	114.0	-380.0
688	254.0	114.0	-380.0
689	0.0	154.0	-380.0
690	40.0	154.0	-380.0
691	80.0	154.0	-380.0
692	84.0	154.0	-380.0
693	94.0	154.0	-380.0
694	114.0	154.0	-380.0
695	154.0	154.0	-380.0
696	254.0	154.0	-380.0
697	0.0	254.0	-380.0
698	40.0	254.0	-380.0
699	80.0	254.0	-380.0
700	84.0	254.0	-380.0
701	94.0	254.0	-380.0
702	114.0	254.0	-380.0
703	154.0	254.0	-380.0
704	254.0	254.0	-380.0
705	0.0	0.0	-300.0

706	40.0	0.0	-300.0
707	80.0	0.0	-300.0
708	84.0	0.0	-300.0
709	94.0	0.0	-300.0
710	114.0	0.0	-300.0
711	154.0	0.0	-300.0
712	254.0	0.0	-300.0
713	0.0	40.0	-300.0
714	40.0	40.0	-300.0
715	80.0	40.0	-300.0
716	84.0	40.0	-300.0
717	94.0	40.0	-300.0
718	114.0	40.0	-300.0
719	154.0	40.0	-300.0
720	254.0	40.0	-300.0
721	0.0	80.0	-300.0
722	40.0	80.0	-300.0
723	80.0	80.0	-300.0
724	84.0	80.0	-300.0
725	94.0	80.0	-300.0
726	114.0	80.0	-300.0
727	154.0	80.0	-300.0
728	254.0	80.0	-300.0
729	0.0	84.0	-300.0
730	40.0	84.0	-300.0
731	80.0	84.0	-300.0
732	84.0	84.0	-300.0
733	94.0	84.0	-300.0
734	114.0	84.0	-300.0
735	154.0	84.0	-300.0
736	254.0	84.0	-300.0
737	0.0	94.0	-300.0
738	40.0	94.0	-300.0
739	80.0	94.0	-300.0
740	84.0	94.0	-300.0
741	94.0	94.0	-300.0
742	114.0	94.0	-300.0
743	154.0	94.0	-300.0
744	254.0	94.0	-300.0
745	0.0	114.0	-300.0
746	40.0	114.0	-300.0
747	80.0	114.0	-300.0
748	84.0	114.0	-300.0
749	94.0	114.0	-300.0
750	114.0	114.0	-300.0
751	154.0	114.0	-300.0
752	254.0	114.0	-300.0
753	0.0	154.0	-300.0
754	40.0	154.0	-300.0
755	80.0	154.0	-300.0
756	84.0	154.0	-300.0

757	94.0	154.0	-300.0
758	114.0	154.0	-300.0
759	154.0	154.0	-300.0
760	254.0	154.0	-300.0
761	0.0	254.0	-300.0
762	40.0	254.0	-300.0
763	80.0	254.0	-300.0
764	84.0	254.0	-300.0
765	94.0	254.0	-300.0
766	114.0	254.0	-300.0
767	154.0	254.0	-300.0
768	254.0	254.0	-300.0
769	0.0	0.0	-150.0
770	40.0	0.0	-150.0
771	80.0	0.0	-150.0
772	84.0	0.0	-150.0
773	94.0	0.0	-150.0
774	114.0	0.0	-150.0
775	154.0	0.0	-150.0
776	254.0	0.0	-150.0
777	0.0	40.0	-150.0
778	40.0	40.0	-150.0
779	80.0	40.0	-150.0
780	84.0	40.0	-150.0
781	94.0	40.0	-150.0
782	114.0	40.0	-150.0
783	154.0	40.0	-150.0
784	254.0	40.0	-150.0
785	0.0	80.0	-150.0
786	40.0	80.0	-150.0
787	80.0	80.0	-150.0
788	84.0	80.0	-150.0
789	94.0	80.0	-150.0
790	114.0	80.0	-150.0
791	154.0	80.0	-150.0
792	254.0	80.0	-150.0
793	0.0	84.0	-150.0
794	40.0	84.0	-150.0
795	80.0	84.0	-150.0
796	84.0	84.0	-150.0
797	94.0	84.0	-150.0
798	114.0	84.0	-150.0
799	154.0	84.0	-150.0
800	254.0	84.0	-150.0
801	0.0	94.0	-150.0
802	40.0	94.0	-150.0
803	80.0	94.0	-150.0
804	84.0	94.0	-150.0
805	94.0	94.0	-150.0
806	114.0	94.0	-150.0
807	154.0	94.0	-150.0

808	254.0	94.0	-150.0
809	0.0	114.0	-150.0
810	40.0	114.0	-150.0
811	80.0	114.0	-150.0
812	84.0	114.0	-150.0
813	94.0	114.0	-150.0
814	114.0	114.0	-150.0
815	154.0	114.0	-150.0
816	254.0	114.0	-150.0
817	0.0	154.0	-150.0
818	40.0	154.0	-150.0
819	80.0	154.0	-150.0
820	84.0	154.0	-150.0
821	94.0	154.0	-150.0
822	114.0	154.0	-150.0
823	154.0	154.0	-150.0
824	254.0	154.0	-150.0
825	0.0	254.0	-150.0
826	40.0	254.0	-150.0
827	80.0	254.0	-150.0
828	84.0	254.0	-150.0
829	94.0	254.0	-150.0
830	114.0	254.0	-150.0
831	154.0	254.0	-150.0
832	254.0	254.0	-150.0
833	0.0	0.0	0.0
834	40.0	0.0	0.0
835	80.0	0.0	0.0
836	84.0	0.0	0.0
837	94.0	0.0	0.0
838	114.0	0.0	0.0
839	154.0	0.0	0.0
840	254.0	0.0	0.0
841	0.0	40.0	0.0
842	40.0	40.0	0.0
843	80.0	40.0	0.0
844	84.0	40.0	0.0
845	94.0	40.0	0.0
846	114.0	40.0	0.0
847	154.0	40.0	0.0
848	254.0	40.0	0.0
849	0.0	80.0	0.0
850	40.0	80.0	0.0
851	80.0	80.0	0.0
852	84.0	80.0	0.0
853	94.0	80.0	0.0
854	114.0	80.0	0.0
855	154.0	80.0	0.0
856	254.0	80.0	0.0
857	0.0	84.0	0.0
858	40.0	84.0	0.0

```
     859    80.0    84.0     0.0
     860    84.0    84.0     0.0
     861    94.0    84.0     0.0
     862   114.0    84.0     0.0
     863   154.0    84.0     0.0
     864   254.0    84.0     0.0
     865     0.0    94.0     0.0
     866    40.0    94.0     0.0
     867    80.0    94.0     0.0
     868    84.0    94.0     0.0
     869    94.0    94.0     0.0
     870   114.0    94.0     0.0
     871   154.0    94.0     0.0
     872   254.0    94.0     0.0
     873     0.0   114.0     0.0
     874    40.0   114.0     0.0
     875    80.0   114.0     0.0
     876    84.0   114.0     0.0
     877    94.0   114.0     0.0
     878   114.0   114.0     0.0
     879   154.0   114.0     0.0
     880   254.0   114.0     0.0
     881     0.0   154.0     0.0
     882    40.0   154.0     0.0
     883    80.0   154.0     0.0
     884    84.0   154.0     0.0
     885    94.0   154.0     0.0
     886   114.0   154.0     0.0
     887   154.0   154.0     0.0
     888   254.0   154.0     0.0
     889     0.0   254.0     0.0
     890    40.0   254.0     0.0
     891    80.0   254.0     0.0
     892    84.0   254.0     0.0
     893    94.0   254.0     0.0
     894   114.0   254.0     0.0
     895   154.0   254.0     0.0
     896   254.0   254.0     0.0
#
# Total number of fixities =  428
# Maximum number of fixities =  464
# Node      Code            x         y         z
     1       111        0.000000  0.000000  0.000000
     2       011        0.000000  0.000000  0.000000
     3       011        0.000000  0.000000  0.000000
     4       011        0.000000  0.000000  0.000000
     5       011        0.000000  0.000000  0.000000
     6       011        0.000000  0.000000  0.000000
     7       011        0.000000  0.000000  0.000000
     8       111        0.000000  0.000000  0.000000
     9       101        0.000000  0.000000  0.000000
```

```
10         001        0.000000  0.000000  0.000000
11         001        0.000000  0.000000  0.000000
12         001        0.000000  0.000000  0.000000
13         001        0.000000  0.000000  0.000000
14         001        0.000000  0.000000  0.000000
15         001        0.000000  0.000000  0.000000
16         101        0.000000  0.000000  0.000000
17         101        0.000000  0.000000  0.000000
18         001        0.000000  0.000000  0.000000
19         001        0.000000  0.000000  0.000000
20         001        0.000000  0.000000  0.000000
21         001        0.000000  0.000000  0.000000
22         001        0.000000  0.000000  0.000000
23         001        0.000000  0.000000  0.000000
24         101        0.000000  0.000000  0.000000
25         101        0.000000  0.000000  0.000000
26         001        0.000000  0.000000  0.000000
27         001        0.000000  0.000000  0.000000
28         001        0.000000  0.000000  0.000000
29         001        0.000000  0.000000  0.000000
30         001        0.000000  0.000000  0.000000
31         001        0.000000  0.000000  0.000000
32         101        0.000000  0.000000  0.000000
33         101        0.000000  0.000000  0.000000
34         001        0.000000  0.000000  0.000000
35         001        0.000000  0.000000  0.000000
36         001        0.000000  0.000000  0.000000
37         001        0.000000  0.000000  0.000000
38         001        0.000000  0.000000  0.000000
39         001        0.000000  0.000000  0.000000
40         101        0.000000  0.000000  0.000000
41         101        0.000000  0.000000  0.000000
42         001        0.000000  0.000000  0.000000
43         001        0.000000  0.000000  0.000000
44         001        0.000000  0.000000  0.000000
45         001        0.000000  0.000000  0.000000
46         001        0.000000  0.000000  0.000000
47         001        0.000000  0.000000  0.000000
48         101        0.000000  0.000000  0.000000
49         101        0.000000  0.000000  0.000000
50         001        0.000000  0.000000  0.000000
51         001        0.000000  0.000000  0.000000
52         001        0.000000  0.000000  0.000000
53         001        0.000000  0.000000  0.000000
54         001        0.000000  0.000000  0.000000
55         001        0.000000  0.000000  0.000000
56         101        0.000000  0.000000  0.000000
57         111        0.000000  0.000000  0.000000
58         011        0.000000  0.000000  0.000000
59         011        0.000000  0.000000  0.000000
60         011        0.000000  0.000000  0.000000
```

```
 61      011       0.000000  0.000000  0.000000
 62      011       0.000000  0.000000  0.000000
 63      011       0.000000  0.000000  0.000000
 64      111       0.000000  0.000000  0.000000
 65      110       0.000000  0.000000  0.000000
 66      010       0.000000  0.000000  0.000000
 67      010       0.000000  0.000000  0.000000
 68      010       0.000000  0.000000  0.000000
 69      010       0.000000  0.000000  0.000000
 70      010       0.000000  0.000000  0.000000
 71      010       0.000000  0.000000  0.000000
 72      110       0.000000  0.000000  0.000000
 73      100       0.000000  0.000000  0.000000
 80      100       0.000000  0.000000  0.000000
 81      100       0.000000  0.000000  0.000000
 88      100       0.000000  0.000000  0.000000
 89      100       0.000000  0.000000  0.000000
 96      100       0.000000  0.000000  0.000000
 97      100       0.000000  0.000000  0.000000
104      100       0.000000  0.000000  0.000000
105      100       0.000000  0.000000  0.000000
112      100       0.000000  0.000000  0.000000
113      100       0.000000  0.000000  0.000000
120      100       0.000000  0.000000  0.000000
121      110       0.000000  0.000000  0.000000
122      010       0.000000  0.000000  0.000000
123      010       0.000000  0.000000  0.000000
124      010       0.000000  0.000000  0.000000
125      010       0.000000  0.000000  0.000000
126      010       0.000000  0.000000  0.000000
127      010       0.000000  0.000000  0.000000
128      110       0.000000  0.000000  0.000000
129      110       0.000000  0.000000  0.000000
130      010       0.000000  0.000000  0.000000
131      010       0.000000  0.000000  0.000000
132      010       0.000000  0.000000  0.000000
133      010       0.000000  0.000000  0.000000
134      010       0.000000  0.000000  0.000000
135      010       0.000000  0.000000  0.000000
136      110       0.000000  0.000000  0.000000
137      100       0.000000  0.000000  0.000000
144      100       0.000000  0.000000  0.000000
145      100       0.000000  0.000000  0.000000
152      100       0.000000  0.000000  0.000000
153      100       0.000000  0.000000  0.000000
160      100       0.000000  0.000000  0.000000
161      100       0.000000  0.000000  0.000000
168      100       0.000000  0.000000  0.000000
169      100       0.000000  0.000000  0.000000
176      100       0.000000  0.000000  0.000000
177      100       0.000000  0.000000  0.000000
```

184	100	0.000000	0.000000	0.000000
185	110	0.000000	0.000000	0.000000
186	010	0.000000	0.000000	0.000000
187	010	0.000000	0.000000	0.000000
188	010	0.000000	0.000000	0.000000
189	010	0.000000	0.000000	0.000000
190	010	0.000000	0.000000	0.000000
191	010	0.000000	0.000000	0.000000
192	110	0.000000	0.000000	0.000000
193	110	0.000000	0.000000	0.000000
194	010	0.000000	0.000000	0.000000
195	010	0.000000	0.000000	0.000000
196	010	0.000000	0.000000	0.000000
197	010	0.000000	0.000000	0.000000
198	010	0.000000	0.000000	0.000000
199	010	0.000000	0.000000	0.000000
200	110	0.000000	0.000000	0.000000
201	100	0.000000	0.000000	0.000000
208	100	0.000000	0.000000	0.000000
209	100	0.000000	0.000000	0.000000
216	100	0.000000	0.000000	0.000000
217	100	0.000000	0.000000	0.000000
224	100	0.000000	0.000000	0.000000
225	100	0.000000	0.000000	0.000000
232	100	0.000000	0.000000	0.000000
233	100	0.000000	0.000000	0.000000
240	100	0.000000	0.000000	0.000000
241	100	0.000000	0.000000	0.000000
248	100	0.000000	0.000000	0.000000
249	110	0.000000	0.000000	0.000000
250	010	0.000000	0.000000	0.000000
251	010	0.000000	0.000000	0.000000
252	010	0.000000	0.000000	0.000000
253	010	0.000000	0.000000	0.000000
254	010	0.000000	0.000000	0.000000
255	010	0.000000	0.000000	0.000000
256	110	0.000000	0.000000	0.000000
257	110	0.000000	0.000000	0.000000
258	010	0.000000	0.000000	0.000000
259	010	0.000000	0.000000	0.000000
260	010	0.000000	0.000000	0.000000
261	010	0.000000	0.000000	0.000000
262	010	0.000000	0.000000	0.000000
263	010	0.000000	0.000000	0.000000
264	110	0.000000	0.000000	0.000000
265	100	0.000000	0.000000	0.000000
272	100	0.000000	0.000000	0.000000
273	100	0.000000	0.000000	0.000000
280	100	0.000000	0.000000	0.000000
281	100	0.000000	0.000000	0.000000
288	100	0.000000	0.000000	0.000000

```
289      100      0.000000  0.000000  0.000000
296      100      0.000000  0.000000  0.000000
297      100      0.000000  0.000000  0.000000
304      100      0.000000  0.000000  0.000000
305      100      0.000000  0.000000  0.000000
312      100      0.000000  0.000000  0.000000
313      110      0.000000  0.000000  0.000000
314      010      0.000000  0.000000  0.000000
315      010      0.000000  0.000000  0.000000
316      010      0.000000  0.000000  0.000000
317      010      0.000000  0.000000  0.000000
318      010      0.000000  0.000000  0.000000
319      010      0.000000  0.000000  0.000000
320      110      0.000000  0.000000  0.000000
321      110      0.000000  0.000000  0.000000
322      010      0.000000  0.000000  0.000000
323      010      0.000000  0.000000  0.000000
324      010      0.000000  0.000000  0.000000
325      010      0.000000  0.000000  0.000000
326      010      0.000000  0.000000  0.000000
327      010      0.000000  0.000000  0.000000
328      110      0.000000  0.000000  0.000000
329      100      0.000000  0.000000  0.000000
336      100      0.000000  0.000000  0.000000
337      100      0.000000  0.000000  0.000000
344      100      0.000000  0.000000  0.000000
345      100      0.000000  0.000000  0.000000
352      100      0.000000  0.000000  0.000000
353      100      0.000000  0.000000  0.000000
360      100      0.000000  0.000000  0.000000
361      100      0.000000  0.000000  0.000000
368      100      0.000000  0.000000  0.000000
369      100      0.000000  0.000000  0.000000
376      100      0.000000  0.000000  0.000000
377      110      0.000000  0.000000  0.000000
378      010      0.000000  0.000000  0.000000
379      010      0.000000  0.000000  0.000000
380      010      0.000000  0.000000  0.000000
381      010      0.000000  0.000000  0.000000
382      010      0.000000  0.000000  0.000000
383      010      0.000000  0.000000  0.000000
384      110      0.000000  0.000000  0.000000
385      110      0.000000  0.000000  0.000000
386      010      0.000000  0.000000  0.000000
387      010      0.000000  0.000000  0.000000
388      010      0.000000  0.000000  0.000000
389      010      0.000000  0.000000  0.000000
390      010      0.000000  0.000000  0.000000
391      010      0.000000  0.000000  0.000000
392      110      0.000000  0.000000  0.000000
393      100      0.000000  0.000000  0.000000
```

400	100	0.000000	0.000000	0.000000
401	100	0.000000	0.000000	0.000000
408	100	0.000000	0.000000	0.000000
409	100	0.000000	0.000000	0.000000
416	100	0.000000	0.000000	0.000000
417	100	0.000000	0.000000	0.000000
424	100	0.000000	0.000000	0.000000
425	100	0.000000	0.000000	0.000000
432	100	0.000000	0.000000	0.000000
433	100	0.000000	0.000000	0.000000
440	100	0.000000	0.000000	0.000000
441	110	0.000000	0.000000	0.000000
442	010	0.000000	0.000000	0.000000
443	010	0.000000	0.000000	0.000000
444	010	0.000000	0.000000	0.000000
445	010	0.000000	0.000000	0.000000
446	010	0.000000	0.000000	0.000000
447	010	0.000000	0.000000	0.000000
448	110	0.000000	0.000000	0.000000
449	110	0.000000	0.000000	0.000000
450	010	0.000000	0.000000	0.000000
451	010	0.000000	0.000000	0.000000
452	010	0.000000	0.000000	0.000000
453	010	0.000000	0.000000	0.000000
454	010	0.000000	0.000000	0.000000
455	010	0.000000	0.000000	0.000000
456	110	0.000000	0.000000	0.000000
457	100	0.000000	0.000000	0.000000
464	100	0.000000	0.000000	0.000000
465	100	0.000000	0.000000	0.000000
472	100	0.000000	0.000000	0.000000
473	100	0.000000	0.000000	0.000000
480	100	0.000000	0.000000	0.000000
481	100	0.000000	0.000000	0.000000
488	100	0.000000	0.000000	0.000000
489	100	0.000000	0.000000	0.000000
496	100	0.000000	0.000000	0.000000
497	100	0.000000	0.000000	0.000000
504	100	0.000000	0.000000	0.000000
505	110	0.000000	0.000000	0.000000
506	010	0.000000	0.000000	0.000000
507	010	0.000000	0.000000	0.000000
508	010	0.000000	0.000000	0.000000
509	010	0.000000	0.000000	0.000000
510	010	0.000000	0.000000	0.000000
511	010	0.000000	0.000000	0.000000
512	110	0.000000	0.000000	0.000000
513	110	0.000000	0.000000	0.000000
514	010	0.000000	0.000000	0.000000
515	010	0.000000	0.000000	0.000000
516	010	0.000000	0.000000	0.000000

```
517        010        0.000000  0.000000  0.000000
518        010        0.000000  0.000000  0.000000
519        010        0.000000  0.000000  0.000000
520        110        0.000000  0.000000  0.000000
521        100        0.000000  0.000000  0.000000
528        100        0.000000  0.000000  0.000000
529        100        0.000000  0.000000  0.000000
536        100        0.000000  0.000000  0.000000
537        100        0.000000  0.000000  0.000000
544        100        0.000000  0.000000  0.000000
545        100        0.000000  0.000000  0.000000
552        100        0.000000  0.000000  0.000000
553        100        0.000000  0.000000  0.000000
560        100        0.000000  0.000000  0.000000
561        100        0.000000  0.000000  0.000000
568        100        0.000000  0.000000  0.000000
569        110        0.000000  0.000000  0.000000
570        010        0.000000  0.000000  0.000000
571        010        0.000000  0.000000  0.000000
572        010        0.000000  0.000000  0.000000
573        010        0.000000  0.000000  0.000000
574        010        0.000000  0.000000  0.000000
575        010        0.000000  0.000000  0.000000
576        110        0.000000  0.000000  0.000000
577        110        0.000000  0.000000  0.000000
578        010        0.000000  0.000000  0.000000
579        010        0.000000  0.000000  0.000000
580        010        0.000000  0.000000  0.000000
581        010        0.000000  0.000000  0.000000
582        010        0.000000  0.000000  0.000000
583        010        0.000000  0.000000  0.000000
584        110        0.000000  0.000000  0.000000
585        100        0.000000  0.000000  0.000000
592        100        0.000000  0.000000  0.000000
593        100        0.000000  0.000000  0.000000
600        100        0.000000  0.000000  0.000000
601        100        0.000000  0.000000  0.000000
608        100        0.000000  0.000000  0.000000
609        100        0.000000  0.000000  0.000000
616        100        0.000000  0.000000  0.000000
617        100        0.000000  0.000000  0.000000
624        100        0.000000  0.000000  0.000000
625        100        0.000000  0.000000  0.000000
632        100        0.000000  0.000000  0.000000
633        110        0.000000  0.000000  0.000000
634        010        0.000000  0.000000  0.000000
635        010        0.000000  0.000000  0.000000
636        010        0.000000  0.000000  0.000000
637        010        0.000000  0.000000  0.000000
638        010        0.000000  0.000000  0.000000
639        010        0.000000  0.000000  0.000000
```

```
640     110     0.000000  0.000000  0.000000
641     110     0.000000  0.000000  0.000000
642     010     0.000000  0.000000  0.000000
643     010     0.000000  0.000000  0.000000
644     010     0.000000  0.000000  0.000000
645     010     0.000000  0.000000  0.000000
646     010     0.000000  0.000000  0.000000
647     010     0.000000  0.000000  0.000000
648     110     0.000000  0.000000  0.000000
649     100     0.000000  0.000000  0.000000
656     100     0.000000  0.000000  0.000000
657     100     0.000000  0.000000  0.000000
664     100     0.000000  0.000000  0.000000
665     100     0.000000  0.000000  0.000000
672     100     0.000000  0.000000  0.000000
673     100     0.000000  0.000000  0.000000
680     100     0.000000  0.000000  0.000000
681     100     0.000000  0.000000  0.000000
688     100     0.000000  0.000000  0.000000
689     100     0.000000  0.000000  0.000000
696     100     0.000000  0.000000  0.000000
697     110     0.000000  0.000000  0.000000
698     010     0.000000  0.000000  0.000000
699     010     0.000000  0.000000  0.000000
700     010     0.000000  0.000000  0.000000
701     010     0.000000  0.000000  0.000000
702     010     0.000000  0.000000  0.000000
703     010     0.000000  0.000000  0.000000
704     110     0.000000  0.000000  0.000000
705     110     0.000000  0.000000  0.000000
706     010     0.000000  0.000000  0.000000
707     010     0.000000  0.000000  0.000000
708     010     0.000000  0.000000  0.000000
709     010     0.000000  0.000000  0.000000
710     010     0.000000  0.000000  0.000000
711     010     0.000000  0.000000  0.000000
712     110     0.000000  0.000000  0.000000
713     100     0.000000  0.000000  0.000000
720     100     0.000000  0.000000  0.000000
721     100     0.000000  0.000000  0.000000
728     100     0.000000  0.000000  0.000000
729     100     0.000000  0.000000  0.000000
736     100     0.000000  0.000000  0.000000
737     100     0.000000  0.000000  0.000000
744     100     0.000000  0.000000  0.000000
745     100     0.000000  0.000000  0.000000
752     100     0.000000  0.000000  0.000000
753     100     0.000000  0.000000  0.000000
760     100     0.000000  0.000000  0.000000
761     110     0.000000  0.000000  0.000000
762     010     0.000000  0.000000  0.000000
```

```
763      010      0.000000  0.000000  0.000000
764      010      0.000000  0.000000  0.000000
765      010      0.000000  0.000000  0.000000
766      010      0.000000  0.000000  0.000000
767      010      0.000000  0.000000  0.000000
768      110      0.000000  0.000000  0.000000
769      110      0.000000  0.000000  0.000000
770      010      0.000000  0.000000  0.000000
771      010      0.000000  0.000000  0.000000
772      010      0.000000  0.000000  0.000000
773      010      0.000000  0.000000  0.000000
774      010      0.000000  0.000000  0.000000
775      010      0.000000  0.000000  0.000000
776      110      0.000000  0.000000  0.000000
777      100      0.000000  0.000000  0.000000
784      100      0.000000  0.000000  0.000000
785      100      0.000000  0.000000  0.000000
792      100      0.000000  0.000000  0.000000
793      100      0.000000  0.000000  0.000000
800      100      0.000000  0.000000  0.000000
801      100      0.000000  0.000000  0.000000
808      100      0.000000  0.000000  0.000000
809      100      0.000000  0.000000  0.000000
816      100      0.000000  0.000000  0.000000
817      100      0.000000  0.000000  0.000000
824      100      0.000000  0.000000  0.000000
825      110      0.000000  0.000000  0.000000
826      010      0.000000  0.000000  0.000000
827      010      0.000000  0.000000  0.000000
828      010      0.000000  0.000000  0.000000
829      010      0.000000  0.000000  0.000000
830      010      0.000000  0.000000  0.000000
831      010      0.000000  0.000000  0.000000
832      110      0.000000  0.000000  0.000000
833      110      0.000000  0.000000  0.000000
834      010      0.000000  0.000000  0.000000
835      010      0.000000  0.000000  0.000000
836      010      0.000000  0.000000  0.000000
837      010      0.000000  0.000000  0.000000
838      010      0.000000  0.000000  0.000000
839      010      0.000000  0.000000  0.000000
840      110      0.000000  0.000000  0.000000
841      100      0.000000  0.000000  0.000000
848      100      0.000000  0.000000  0.000000
849      100      0.000000  0.000000  0.000000
856      100      0.000000  0.000000  0.000000
857      100      0.000000  0.000000  0.000000
864      100      0.000000  0.000000  0.000000
865      100      0.000000  0.000000  0.000000
872      100      0.000000  0.000000  0.000000
873      100      0.000000  0.000000  0.000000
```

```
  880       100      0.000000  0.000000  0.000000
  881       100      0.000000  0.000000  0.000000
  888       100      0.000000  0.000000  0.000000
  889       110      0.000000  0.000000  0.000000
  890       010      0.000000  0.000000  0.000000
  891       010      0.000000  0.000000  0.000000
  892       010      0.000000  0.000000  0.000000
  893       010      0.000000  0.000000  0.000000
  894       010      0.000000  0.000000  0.000000
  895       010      0.000000  0.000000  0.000000
  896       110      0.000000  0.000000  0.000000
#
#
# Material value identifier
# Oil sand
 1
# All stresses are in kPa
#   1      2       3        4       5              6    7    8      9
# Young Poisson Dilation Matden Uniax(cohesion) Hard  Phi  Eps  Alpha
  2.2e6  0.3    0.0      2.5    0.0              0.0  30.0  2.0  5.0e-5
#
# overburden (shale and till):
# small cohesion, medium E, medium friction, small hardening
 2
  1.0e6  0.3    0.0      2.5    100.0            1.0  25.0  2.0  5.0e-5
#
# underlying limestone: high cohesion, high E, low friction, zero hardening
 3
  30.0e6 0.3    0.0      2.6    24.0e3           0.0  28.0  2.0  4.0e-5
#
# Initial stress flag
# istres
    1
#
REGIONAL PRINCIPAL STRESS FIELD
#  alpha,   sdz,    A,      B,      C,     E,     hratio,  top,   x-over
   0.0      0.0    0.0    -17.0    0.0    0.0      1.0      0.0    0.0
#
#
# Loading information: a title for each one.
#  igrav, iplod, isurf
     1      0     0
#
GRAVITATIONAL BODY FORCES AND PORE PRESSURE LOADS
  9.81
#
# this is for the initial loading
#
   1
   1.0    0.1    20     0  1  1  1  0  1  0  0  0
                        0  0  1  1  0  1  0  0  0
```

```
#
#CONSTANT SURFACE LOADS
# nsurf,  alpha,    sngaus
#    1     0.0      2
#j,  elem#,    ln1, ln2, ln3, ln4,      sdh1,     sdh2,     sdz
# 1   7         29   30   31   32         00.0      00.0    -20.0e3
#
# load increment parameters (ninc of them)
#
# use for next lnincs increments
# facto, toler, miter, pflags(1..6,1)...Next line pflags(1..6,2)
#
#   this is for the main loading
#
   1
   1.0    0.1    20      0  1  0  0  0  0  0  0  0
                         0  0  1  0  0  0  1  1  1
```

Appendix H

SAMPLE DATA FOR USE WITH SIM3D3P

This data was used for the staggered run described in Chapter VIII.

```
##   Two lines of title material
  ABC1  * CYCLIC STEAM STIMULATION* NO BOTTOM WATER
 10 ACRE SPACING   * 11 M (36 FT) OIL ZONE * 3X3X3 GRID * 150 bpd inj rate
#
# NX, NY, NZ, IHTG, ITEMAX, IESC, NRTIPO, NSWO, NSGO, NTEMP, NTVO, IMBDR
   3    3    3    1 9000    0    1   12   13    1    5, 0
#  underrelaxation factors.
#  URFACI: underrelaxation factor for first URFACN N-R iterations
#  ASOFT.    1: Ev multiplier
#            2: Vr multiplier
#            3: Ev/Vr predictor underrelaxation
#            4: Kfact multiplier (if zero, Kfact not used [==1])
#            5: Pseudo-Cr (if > 0, use this to implicitly couple problem)
#            6: porosity multiplier
#  urfaci, urfacn, asoft(1), (2), (3), (4), (5), (6), User_Cr
   0.0,     0,      30.0,  1.0, 1.0,  1.0, -1.0e6, 1.0   3.0e-6
#  tolerance and iteration limits for MIXED formulation
# portol     mixmxit
  1.0e-4,    3
#  NR iteration tolerances: epsdt, epsdsw, epsdp
#  0.1,  0.05, 0.1  (SMFA used these for dT and dp).
  0.1, 0.05, 0.5
#
#  CWP,      CWT,       COP,     COT,      CR,       HSPO,      HCN,     HCNOB
 .000003   .00049    .000005   .0005     .0001      225.      42.3      35.3
# .000003    .00049    .000005   .0005     3.0e-6     225.      42.3      35.3
# DENMA,  HSPMA,     DIF
 165.     0.20       0.96
#
# TVIST,   VISOT,   VISWT,    VIS_steam
  75.0    1275.0    0.92      0.0095
  90.0     560.0    0.78      0.0102
  150.0    90.0     0.435     0.0115
  275.0    12.5     0.211     0.0127
  575.0    1.29     0.111     0.0138
#
#   DOINI,  DWINI,   MWTO,      ..           ..
  0.1350    3.49    450.0     0.00000277  11546.82
#
# TEMF,     PSURF,          TSURF
  74.      14.7      80.
#
# SET (L) #1
# SWT(L,I),PCWOT(L,I),KRWT(L,I),KROWT(L,I)
  0.20     0.0       0.0        1.0
  0.25     0.0       0.0102     0.769
  0.294    0.0       0.0168     0.7241
  0.357    0.0       0.0275     0.6206
  0.414    0.0       0.0424     0.504
  0.490    0.0       0.0665     0.3714
  0.557    0.0       0.097      0.3029
  0.630   0.0       0.1148    0.1555
  0.673   0.0       0.1259    0.0956
  0.719   0.0       0.1381    0.0576
  0.789   0.0       0.1636    0.0
  1.0     0.0       0.250     0.0
#
# SET (L) #1
# SLT(L,I),PCGOT(L,I),KROGT(L,I),KRGT(L,I)
  0.29    1.5       0.0       0.17
  0.395   1.0563    0.0294    0.112
  0.433   0.8958    0.0461    0.1022
  0.515   0.5493    0.0883    0.0855
  0.569   0.3211    0.1172    0.0761
  0.614   0.1310    0.1433    0.0654
  0.663   -0.0761   0.17640   0.05
  0.719   -0.3127   0.2170    0.0372
  0.75    -0.4437   0.2255    0.0285
  0.805   -0.6761   0.2919    0.0195
  0.85    -0.8662   0.3373    0.0121
  0.899   -1.0732   0.5169    0.0026
  1.00    -1.500    1.0       0.0
#
# TEMPT, SWCT, SGCT, SORWT, SORGT, SGRT, KRWROT, KROCWT, KRGROT, PCWT, PCGT
  74.0 0.2     0.0    0.211  0.09    0.0   0.1636 1.0    0.17   1.0    1.0
#
# OSRMN,    QOLIM,      TOTIM
   0.1      10.0      1100.0
# DELX(I)
130.0      130.0     130.0
# DELY(I)
130.0      130.0     130.0
# DELZ(I)
 12.0       12.0      12.0
# DIPX,    DIPY,    ZINIC
  0.0      0.0      1800.0
#
#  KX,       KY,       KZ,       PHI  (For each layer).
   2.0       2.0       2.0       0.31
   2.0       2.0       2.0       0.31
   2.0       2.0       2.0       0.31
#
#   I1,  I2,  J1,  J2,  K1,  K2
    1    3    1    3    1    3
# ZINIC,  PINIC,    ZWOC,     ZGOC,       PCWOC,   PCGOC
1800.       150.      0.0       0.0       -100.4     -100.
    0    0    0    0    0    0                                BLANK11
    0    0    0    0    0    0                                BLANK12
    0    0    0    0    0    0                                BLANK13
    0  0  0    0.0,   0.0,      0.0,      0.0,      0.0                 BLANK
#  I1,  I2,  J1,  J2,  K1,  K2, IS1
    0    0    0    0    0    0    1
```

```
#    LAYER,  PX,       TX,        SWX,      SGX
         1  200.       74.       0.23      0.0                 INIT.
         2  200.       74.       0.23      0.0                 INIT.
         3  200.       74.       0.23      0.0                 INIT.
#
# Well Data info from below here ----------------------------------------------
#
# TCAMB,    DT,   DT1,    DSMX,     DPMX,    DTPMX,   DTMAX,      DTMIN
    30.     0.0   0.5     0.01       20.       20.      20.        0.005
#  I01, I02
    1    1
#  NR iteration tolerances: epsdt, epsdsw, epsdp
  0.1, 0.05, 0.5
# INDQ,IPOZO,IQVAP,MNITN,MXITN
    0    1    0    2   10
#  I,   IP,  JP, KP1, KP2, INDP1,INDP2,ITASA,KBOMB
    1    1    1    1    3    0    2    0    1
# WI,     PWF,      PINY,      X      TINY
 150.      0.        500.     0.7    0.0
    0
#
    35.     0.0   0.5     0.01       20.       20.      20.        0.005
    1    1
#  NR iteration tolerances: epsdt, epsdsw, epsdp
  0.1, 0.05, 0.5
    0    1    0    2   10
    1    1    1    1    3    0    2    0    1
  0.      0.          500.     0.7    0.0
    0
#
   125.     0.0   0.0     0.01       20.       20.      20.        0.005
    1    1
#  NR iteration tolerances: epsdt, epsdsw, epsdp
  0.1, 0.05, 1.0
    0    1    0    2   10
    1    1    1    1    3    0    1    0    1
  2.5     0.          500.     0.7    0.0
    0
#
   155.     0.0   0.0     0.01       20.       20.      20.        0.005
    1    1
#  NR iteration tolerances: epsdt, epsdsw, epsdp
  0.1, 0.05, 1.0
    0    1    0    2   10
    1    1    1    1    3    0    2    0    1
 150.      0.         500.     0.7    0.0
    0
#
   160.     0.0   0.0     0.01       20.       20.      20.        0.005
    1    1
#  NR iteration tolerances: epsdt, epsdsw, epsdp
```

```
  0.1, 0.05, 1.0
    0    1    0    2   10
    1    1    1    1    3    0    2    0    1
   0.       0.         500.      0.7    0.0
    0
#
   250.      0.0   0.0     0.01      20.        20.        20.        0.005
    1    1
#  NR iteration tolerances: epsdt, epsdsw, epsdp
  0.1, 0.05, 1.0
    0    1    0    2   10
    1    1    1    1    3    0    1    0    1
  2.5       0.         500.      0.7    0.0
    0
#
   280.      0.0   0.0     0.01      20.        20.        20.        0.005
    1    1
#  NR iteration tolerances: epsdt, epsdsw, epsdp
  0.1, 0.05, 1.0
    0    1    0    2   10
    1    1    1    1    3    0    2    0    1
 150.       0.         500.      0.7    0.0
    0
#
   285.      0.0   0.0     0.01      20.        20.        20.        0.005
    1    1
#  NR iteration tolerances: epsdt, epsdsw, epsdp
  0.1, 0.05, 1.0
    0    1    0    2   10
    1    1    1    1    3    0    2    0    1
   0.       0.         500.      0.7    0.0
    0
#
   375.      0.0   0.0     0.01      20.        20.        20.        0.005
    1    1
#  NR iteration tolerances: epsdt, epsdsw, epsdp
  0.1, 0.05, 1.0
    0    1    0    2   10
    1    1    1    1    3    0    1    0    1
  2.5       0.         500.      0.7    0.0
    0
#
   405.      0.0   0.0     0.01      20.        20.        20.        0.005
    1    1
#  NR iteration tolerances: epsdt, epsdsw, epsdp
  0.1, 0.05, 1.0
    0    1    0    2   10
    1    1    1    1    3    0    2    0    1
 150.       0.         500.      0.7    0.0
    0
#
```

Appendix I

USER MANUAL FOR FESPS

This appendix is a user manual for preparing data sets to be used with FESPS. Two data files are used in a normal run, but one of these is supplied by the preprocessor RENUM. A restart file might also be used if the simulation is to restart from a given time. Running RENUM is part of the data preparation for FESPS, so a separate section is devoted to setting up the input file for RENUM. Keywords are used as much as possible to avoid cryptic controls. A schematic of how input is provided to FESPS is given in Figure I–1.

Most of the data are read in as free format and this must be assumed unless stated otherwise. Thus, all numbers must be separated by spaces and, or, commas. All text must start in column number one of the file: this includes text data and keywords. Comment lines are permitted anywhere in the files with a few exceptions. The terms "card" and "line" are used interchangeably. A line is a line of data terminated with a carriage return. Variable names and other text that is taken from the program appear in `typewriter type`. The type of a variable will usually be indicated in round brackets (`like this`). The variable type `character` will be abbreviated as `char`. Keywords will be distinguished in the text by the use of italics. A range is indicated in the style of Pascal by joining the end-points together by two dots, e.g., `1..n`. Symbolic data sets outlining the input data file structures are given at the end of each section. Use these while going through the explanations. Example problems contain actual data.

I. 1 *Preprocessor* RENUM

I. 1. 1 *Description*

RENUM generates a steering array for an optimal numbering of the degrees of freedom dependent on the connectivity array. It does not renumber the nodes, but reumbers the degrees of freedom instead. Typically, there are four to seven degrees of freedom per node — S_w, S_o, p_w, T and three displacements. It also provides a template for the solver, based on this information. The user neither sees nor cares about the steering array and permutation index. They only matter while the coefficient matrix is being filled, and when the unknown variables have been solved for and the program extracts the degrees of freedom in groups of the degree of freedom type — S_w, S_o, p_w, T, etc. — which are now appropriately ordered by node or element.

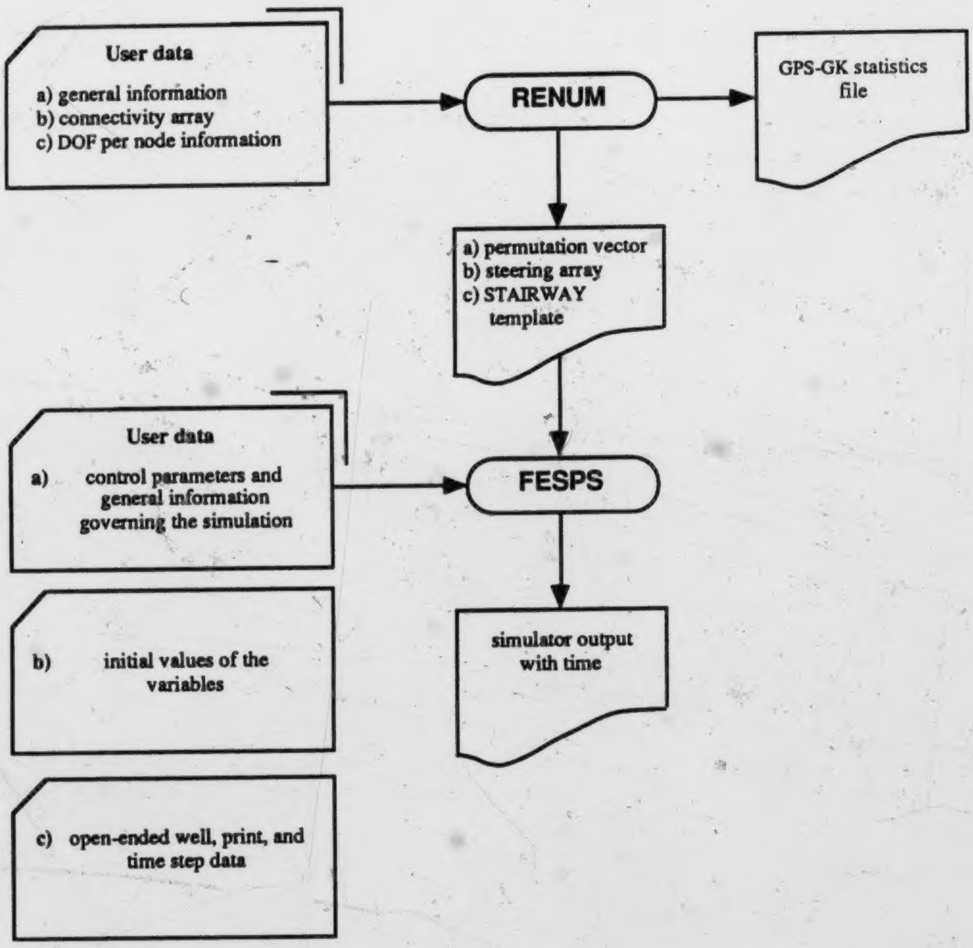

User data
a) general information
b) connectivity array
c) DOF per node information
RENUM
GPS-GK statistics file
a) permutation vector
b) steering array
c) STAIRWAY template
User data
a) control parameters and general information governing the simulation
FESPS
simulator output with time
b) initial values of the variables
c) open-ended well, print, and time step data

Figure I–1

Outline of user input to run FESPS.

I.1.2 *Data for* RENUM

The data are read in three groups:

a) general information;

b) the connectivity array for elements and nodes;

c) the DOF steering array data.

a) There are four cards with the general information. The first card defines the comment character (`char*1`), which will be the first character appearing in the file. A typical choice would be a '#' or a '*'. The second card must contain the key (`char*4`) for the type of minimization to be performed: bandwidth or profile minimization. Write these explicitly as *bandwidth* or *profile minimization* in the file from column one. The third card contains the title of the run (`integer*80`). The fourth card contains the defining parameters of the problem. The four cards are:

```
# a sample comment character
a title line
profile minimization
nelem, npts, nnodes
```

`nelem` is the number of elements in the mesh; `npts` is the number of nodes or points in the mesh; `nnodes` is the maximum number of nodes per element, which is currently eight in FESPS.

b) There are `nelem` cards for the connectivity array which defines how the mesh is divided into elements—one card for each element definition:

```
k, node1, node2, node3, node4, node5, node6, node7, node8
```

`k` is the element number, and the `node1`,... are the local node numbers expressed in the global node numbering scheme. Note that the global scheme is the one chosen by the user, and has nothing to do with the optimal ordering schemes. The values of `node7` and `node8` are set to zero for L6 elements. The elements must appear in ascending order to help the user detect missing data.

c) The degree of freedom numbering section indicates which degrees of freedom are active at which node. A numbering code for the degrees of freedom must be implemented: $1 = S_w$, $2 = So$, $3 = p_w$, $4 = T$, $5 = u$, $6 = v$, $7 = w$. If a given degree of freedom will potentially be on and off during a run (e.g. when a well node changes from a rate to a pressure specification), leave it in the set. The data are entered in groups. The first card indicates the number of groups, `ngroups`, and the maximum degrees of freedom allowed per node, `mxdfpn`, both type `integer`. The number of groups is for the user's benefit and the rest of the data are keyword-controlled. The keyword for a group (`char*3`) is *group*, and exceptions to the group appear before the next group. The input of data are terminated by the keyword *end*. Each keyword must appear on its own card. A group of nodes are initialized with two cards:

i) `node1, node2, ndofpn`

`node1` and `node2` respectively are the start and end nodes of the range of the group; `ndofpn` is the number of degrees of freedom per node in this range.

ii) `whichdof(1..ndofpn)`

This is the list of the degrees of freedom in ascending order.

If the user wishes to specify exceptions to the group definition, he can follow the previous group cards with one card for each exception. The exceptions are terminated by the signal for a new group or the end keyword. Each exception is defined by a card containing the following information:

`nodeno, nodinf(1..mxdfpn)`

`nodeno` is the node number to be changed, and must lie between `node1` and `node2`; `nodinf` is the list of degrees of freedom for the exception; `mxdfpn` digits must appear—pad non-existent degrees of freedom with zeros. The exceptions must appear in ascending order of `nodeno` and must be nodes that fall within the range of the group. Anything else will generate an error. This data could be all provided within separate groups, but the exceptions are more compact to write.

I.2 *Symbolic data set for* RENUM

```
# first character in file is comment character
# wherever it appears in the first column it defines a comment line
# bandwidth minimization
# (we chose profile minimization)
profile minimization
this line is the title of the run
nelem, npts, nnodes
# now need nelem cards for the connectivity array, k=1..nelem
k, node1, node2, node3, node4, node5, node6, node7, node8
# indicate the expected number of groups defining the dof
# and the maximum permitted number of dof per node
ngroups, mxdfpn
# this is a group---indicated by keyword
group
node1, node2, ndofpn
whichdof(1..ndofpn)
# any exceptions follow, terminated by any keyword
nodeno, nodeinf(1..mxdfpn)
nodeno, nodeinf(1..mxdfpn)
# N.B. pad out nodeinf with zeros for non-existent DOF
# because each node must have an equal no. of DOF
# next group
group
node1, node2, ndofpn
whichdof(1..ndofpn)
# no exceptions this time
# terminate the input
end of input
```

I. 3 *Main program* FESPS

FESPS requires two input files—one supplied by the RENUM preprocessor and one supplied by the user. This section describes how to set up the file supplied by the user. It must be consistent with the data used to run the preprocessor RENUM. To enforce this, the connectivity array used by RENUM is echoed to the file it supplies to FESPS. The user does not supply it directly. If the user wishes to alter the connectivity array, he is forced to rerun RENUM with a new input file. The user should never alter the output of RENUM, except to add comments. The output is written to a text file only for the convenience of transfer between different computers and for the user to see the statistics of the minimization.

The user data are supplied in three main groups with partial use of keywords:

a) control parameters and general information governing the simulation;

b) initial values of the variables;

c) well, print, and time step data (open-ended).

I. 3. 1 *Control parameters and general information*

The first character in the file (line one, column one) defines the comment character. A typical choice would be a '#' or a '*'. Thus the first line is always a comment. Various control data follow the comment definition. Three cards appear first: a title for the simulation up to 80 characters long (`char*80`) and basic information for the run. Thus the first four cards in the file will look like:

```
# a sample comment character
a title line
strtstep, npts, nelem, mxdofpn, totdof, ngaus
idof(1), idof(2), idof(3), idof(4), idof(5), idof(6), idof(7)
```

`strtstep` is the time step at which to start the simulation. A new simulation requires `strtstep` = 0. A restart record file must be supplied if the simulation is to start at any positive value. A negative value causes the data to be checked but no simulation to be run. `npts` is the number of nodes in the mesh; `nelem` is the number of elements in the mesh; `maxdofpn` is the maximum number of degrees of freedom permitted per node; `totdof` is the total number of degrees of freedom in the problem; `ngaus` (14) is the number of Gauss quadrature points chosen to evaluate the integrals. Some of these values can be obtained from the output from RENUM, and are designed to check the input files against each other. All of them are integers. If the problem definition is larger than the compilation parameters, the program will abort at this time. If the problem must still be run, those parts of the source code containing parameter values will have to be recompiled. The problem definition must also agree with the values in the RENUM output file. On the fourth card, enter information about which degrees of freedom are active for this run by typing its value (1...7) or a zero if it is not. E.g., `1,2,3,0,0,0,0` implies an isothermal run. The number of degrees of freedom recognized is controlled by **`mxdofpn`**

The user's choice of units must now be defined using a set of key numbers (`integer` and keywords (**`char*10`**). These choices define which units are used for subsequent input and output. The first card must have one of ***input units*** or ***default units*** starting in the first column. If the default units are chosen, the SI option will be chosen for input and output. Otherwise, at least one input unit must be specified using key numbers and keywords. Up to sixteen units can be defined. Any units not mentioned in this table will be derived from this set. The variable is identified by a number and its units by a keyword. The keyword must appear on the next uncommented card and start in the first column. A keynumber of zero terminates the current set of unit definitions.

The keywords are self-explanatory, and are listed in Table I–1. Any units not specified by the user will be set to the default SI unit. The input units must be followed by the output units specifications. How they will be defined is controlled by the keywords ***same units as input***, or ***output units*** if they are to be specified. These must begin in the first column. It is helpful to identify the key number by appending a description to the line, as only one field is read from the line. For example,

```
input units
keynumber - description
keyword
same units as input
```

If output units are to be specified use the same key numbers and key words as given in Table I–1. It is helpful to identify the variables in the input file using comments. Note that all Btu used are International Table values.

Several cards of control data now follow. First are the controls for time step growth and saturation overshoot. Typical values are given in parenthesis during the explanation. See also the paper by Grabowski et al, 1979.

```
pnorm, tnorm, satnorm, dispnorm, dampdt, satcon, theta
```

`pnorm` (0.5 MPa), **`tnorm`** (30 K), **`satnorm`** (0.2) , and **`dispnorm`** (1 mm) are the norms controlling time step growth; **`dampdt`** (1.25) is the growth relaxation factor; **`theta`** (0.5) is the collocation point distance for the time discretization; **`satcon`** (10^{-5}) is the saturation overshoot tolerance above which

key number	*variables*	*valid keywords for units*
0	end of current units definition	
1	time	`secs`
		`days`
		`hours`
		`mins`
2	pressure	`Pa`
		`kPa`
		`MPa`
		`bar`
		`atm`
		`psia`
3	temperature	`K`
		`deg C`
		`deg R`
		`deg F`
4	absolute permeabilities	`m2`
		`_um2`
		`darcy`
		`md`
5	length	`m`
		`cm`
		`feet`
6	densities	`kg/m3`
		`g/cm3`
		`lbm/USgal`
		`lbm/ft3`
7	viscosities	`Pa.s`
		`mPa.s`
		`cp`
8	volumes	`m3`
		`dm3`
		`USgal`
		`ft3`
		`bbl`

Table I–1

Key numbers and words for user-specified units.

key number	*variables*	*valid keywords for units*
9	energy	`J`
		`kJ`
		`MJ`
		`Btu`
10	mass	`kg`
		`g`
		`lbm`
11	thermal conductivity	`W/(m.K)`
		`kW/(m.K)`
		`Btu/h-ft-F`
12	well volumes	`m3`
		`dm3`
		`USgal`
		`ft3`
		`bbl`
13	well rates	`m3/s`
		`dm3/s`
		`m3/d`
		`USgal/d`
		`ft3/d`
		`bbl/d`
14	displacements	`m`
		`mm`
		`feet`
		`inches`
15	stresses	`Pa`
		`MPa`
		`psia`
16	solution gas ratio	`sm3/sm3`
		`vol/vol`
		`scf/bbl`
17	well energy rates	`J/s`
		`MJ/d`
		`kJ/d`
		`MJ/s`
		`kJ/s`
		`Btu/d`

Table I–1. Key numbers and words for user-specified units (cont.).

a time step must be repeated. `satnorm` can take a maximum value of 0.3, which permits a change of up to three times the previous value.

A card with the times for the simulation comes next.

`endtime, dtmax, rstrtint, rstflag`

`endtime` is the end time of the simulation (`real`); `dtmax` is the maximum number of time steps (`real`); `rstrtint` is the interval in time steps between writing restart records (`integer`) > 0; `rstflag` indicates whether or not to write a restart record at the same time as detailed output is printed (`integer`). A value of one for `rstflag` enforces the writing of the record.

Four cards follow with data in the same order as described next, with typical values following in parenthesis:

i) `tols(1..9)`

These are the tolerances (`real`) for:

1. pressure convergence for water density calculations (1.0 kPa);
2. convergence for z-factor calculations (10^{-5});
3. the ϵ parameter for S_g (10^{-4});
4. the convergence of the Newton-Raphson saturation solution (0.002);
5. the convergence of the Newton-Raphson pressure solution (1 kPa);
6. the convergence of the Newton-Raphson temperature solution (0.5 K);
7. the convergence of the Newton-Raphson displacement solution (0.01 mm);
8. the convergence of the function residual for Newton-Raphson method (0.001).
9. the relative convergence of each variable in the linear solver compared to its Newton-Raphson convergence tolerance (0.001).

ii) `shifts(1..5)`

These are the shifts (`real`) for the formation of numerical derivatives:

1. derivatives in the properties routines with respect to pressure (10 kPa);
2. derivatives in the properties routines with respect to temperature (1 K);
3. derivatives in the Jacobian (stiffness) matrix with respect to saturations (10^{-5});
4. derivatives in the Jacobian (stiffness) matrix with respect to p_w (10 Pa);
5. derivatives in the Jacobian (stiffness) matrix with respect to T (10^{-3}).

iii) `iter(1..4)`

These are the iteration controls (`integer`):

1. maximum iterations for z-factor calculations (10);

2. maximum iterations for the Newton-Raphson scheme (5);
3. maximum number of repeated time steps (5).
4. maximum number of linear solver iterations (0).

iv) `perturb(1..4)`

These are the perturbation values of S_w, S_o, p_w, and T, respectively, expressed in fraction pore volume, Pa, or K, per day.

Note that the shifts should be three or four orders of magnitude smaller than the tolerances. The number of iterations requested for the linear solver DSTAIR controls the method of solution. An entry of zero iterations indicates that no iterative improvement is to be found nor is a residual to be calculated. Requesting one or more iterations indicates that the residual is to be found and convergence determined using the product of the relative tolerance and the Newton-Raphson convergence tolerance.

The control parameters section is completed by specifying the level of debugging required. This is all keyword controlled (`char*10`), and they are self-explanatory. Note that the section must be terminated with the end keyword. The current list of possible options is given in Table I–2.

I. 3. 2 *Initial values of variables*

After the various controls have been set, the file produced by RENUM is read in. Then FESPS continues to read from the user input file and initializes the variables. This section is keyword controlled (`char*5`) and most of the data are read by a common subroutine **RESIN**. This subroutine has its own keyword controls (`char*3`) and allows the mesh data to be read in groups of individual data or a single value for a range, or any combination of these. Data are read for the current variable until an end key is provided. The keywords in **RESIN** are:

one
ind
end

Only three characters are significant. These are abbreviations for *one* value for entire range, *ind*ividual data for range, and *end* of data. The *end* keyword helps avoid confusion. Data are entered on two lines with the key and range of mesh nodes:

```
key
start, end, value
```

The `value` is only necessary when `key` is *one* and then this value is assigned to the variable for the

keyword	*description*
`end_debugging`	end of debugging information
`bitmakr`	bit vectors from `bitmakr`
`gauss info`	gauss points and weightings from `gaussq`
`gauss wts`	gauss weightings only from `gaussq`
`tri coords`	triplicate coordinate sets and `bitprsm9` structures
`shape fcns`	shape functions from `shapesl8`
`shape drv`	shape function derivatives from `shapesl8`
`jac shape1`	large information set from transformations of shape function derivatives in `derivsl8`
`jac shape2`	small information set from transformations of `derivsl8`
`jac wtgs1`	large information set from transformations of weighting function derivatives
`jac wtgs2`	small information set from transformations of weighting function derivatives
`wtg w`	L8 and L6 weighting parameters for water
`wtg o&g`	L8 and L6 weighting parameters for oil and gas
`potl w`	potentials for water for weighting functions
`potl o&g`	potentials for oil and gas for weighting functions
`wtg fcn w`	weighting functions for water
`wtg fcn og`	weighting functions for oil and gas
`wtg drv w`	weighting function derivatives for water
`wtg drv og`	weighting function derivatives for oil and gas

Table I–2

Key words for debugging options.

keyword	*description*
`int S1`	integral S1 (SN1N1N)
`int S2`	integral S2 (SN1DN1DW)
`int S3`	integral S3 (SW1DN)
`int S4`	integral S4 (SN1N)
`int S5`	integral S5 (SW1N)
`int S6`	integral S6 (SN1DN1W)
`int S7`	integral S7 (SN1DN2DN)
`test input`	echo of input
`input variables`	immediate echo of input
`well inputs`	well input information
`z-factors`	data from *z*-factor calculations
`subroutine announcements`	give announcement of entry into each subroutine
`well calculations`	data from well calculations
`pointer integers for wells`	list vector pointers for well parameter storage
`DSTAIR printing`	print info. during DSTAIR calculations
`DSTAIR iterative improvement`	info. on iterative improvement
`DSTAIR relative error`	monitor relative error
`DSTAIR resource usage`	monitor usage of computer resources

Table I-2. Key words for debugging options (cont.).

inclusive range of `start` to `end`, both integers. If `key` is *ind* then one value must be provided for each node in the inclusive range. The values are read in free format extending over as many lines as required. No comment lines are permitted within the data, though. If `key` signals the end of the data, no range is necessary.

Exceptions to this form of input are (1) the coordinates of the nodal locations, (2) the initial pressures, (3) the initial bubble point pressures, (4) the convective term multipliers, and (5) the capillary pressure data.

(1) The nodal coordinates are read in for each node with its (x, y, z) coordinates. The nodes must be entered in ascending order. This is enforced to help prevent one from accidentally missing nodes. A reference depth for the z-coordinate must be entered first.

```
reference depth
node, x, y, z
```

Note that the z-direction is positive upwards.

(2) The pressures can either be initialized using equilibrium from a given datum, or be specified individually as in the previous description of `RESIN`. If equilibrium initialization is chosen, the following data must be provided:

```
datum, pdatum, tdatum, goc, woc
```

`datum` is the depth of the datum measured positive downwards; likewise `goc` and `woc` are the depths to the gas-oil and water-oil contacts; `pdatum` and `tdatum` are the pressure and temperature respectively at the datum. If there is no actual GOC or WOC in the reservoir, locate them somewhere outside of the interval of interest. Note that although the program works with a positive-upwards z-coordinate, the points are given as positive depths from the surface.

(3) The initial bubble point pressures of the oil can be set equal to the initial node pressure or equal to some single value. In the latter case it is assumed that the entire domain is undersaturated. Where permeabilities are set equal to the ratio of the permeability in another direction, the multiplying factor for the new set of permeabilities is put on the line following the keyword.

(4) The convective term multipliers. These terms are used for debugging, and are not required. They are entered as required for each of the equations — oil, water, gas, and energy — four terms per equation, except for 24 for the energy equation.

(5) The capillary pressure terms. There are default values as described in the section on rock and fluid properties, or these terms can be entered as:

```
Swi, Sor, epsilon, a, b, c, d, m
```

The keywords specifying the variable to be initialized and the method to use are given in Table I–3. Unless otherwise stated, all keywords start in column one of the data file and are in lower

keyword	*description*
`nodes`	enter nodal locations
`temperature`	enter temperature data
`pressure`	enter temperature data
`equilibrium`	if it follows the pressure keyword, use equilibrium initialization for pressures. If it follows the bubble point pressure keyword, use equilibrium initialization for the initial bubble point pressures
`sato`	enter oil saturation data
`satw`	enter water saturation data
`porosity`	enter porosity data
`bpp`	enter bubble point pressure information
`specified`	if it follows bpp keyword, set the initial bubble point pressure to the following specified value
`cmod`	enter convective term multipliers
`capi`	enter capillary pressure terms
`x-permeability`	enter x-permeability data
`y-permeability`	enter y-permeability data
`z-permeability`	enter z-permeability data
`kx=y`	set $k_x = f \times k_y$
`kx=y`	set $k_x = f \times k_z$
`ky=x`	set $k_y = f \times k_x$
`ky=z`	set $k_y = f \times k_z$
`kz=x`	set $k_z = f \times k_x$
`kz=y`	set $k_z = f \times k_y$
`end`	end of data

Table I–3

Key words for initialization of variables.

case type. Data are read until the end keyword is encountered. The end of this data section is also the end of the static definitions.

I. 3. 3 *Well, print, and control data*

The tail of the user data file is composed of information governing wells, time of printing, and time step controls. As with the other sections, it is keyword controlled. The data will be used only when the specified simulation time is reached. If the current time is greater than or equal to the specified time, the data are read to the next keyword, where the time is again checked. At the start of a simulation, the current time is zero. When the current time is less than the specified time, the remaining data are not read until the simulation time reaches the specified time. Thus new well, print, or control data can be added to a restarted simulation. Simulations with common first parts in time need not be run from the start in this case. Situations might include difficulty converging to a solution at a particular time, or altering a well specification. The data can be added in any order of the three main groups, but it will initially only be read to the first group where the time it is to be used is greater than the current simulation time. The rest of the data will only be read as the simulation time becomes equal to the time to be used: `nextinfo`.

A group of data always begins with two cards:

```
datatype
nextinfo
```

`datatype` is a keyword for the type of data: *well*, *print*, *control*, or an end sequence. `nextinfo` is the activation time in the time input units when the group of data is to be used. Groups of data can be appended as required. The four data types are discussed in turn.

```
--------
```

Four or more dashes indicate the end of all data. It is not necessary but can be considered a statement of data termination.

`well data`

The actual data follow the activation time. At least one card is necessary to define the type of well, and several more to define the geometry.

i) `wnum, type, spec, rw, perf`

`wnum` is the number identifying the well, an integer ranging from 1 to `maxwells`. `type` is the type of well (`integer`), and must be one of

1. total fluid pressure specified production well;
2. bitumen rate specified production well;
3. (gas rate specified production well;)
4. water rate specified production well;
5. water and gas pressure specified injection well;
6. water and gas rate specified injection well;
9. shut in well;

Numbers other than those listed above are invalid; `spec` is the specified pressure or rate of the well, depending on the choice of `type`; `rw` is the wellbore radius; `perf` indicates whether the well will have a new geometry. If the well is being shut in, no further data are required. If it is an injection well, the mass fraction of water in the injected stream and the steam quality must be provided (`real`):

`wsplit(1), wsplit(2)`

The mass fraction of the water is 1.0000 if no gas is being co-injected. The well name (`char*64`) follows on the next line:

`wellid`

Several constraints (`real`) must now be specified on one line:

ii) `con(1), con(2), con(3), con(4), con(5), con(6), con(7)`

where the indices are

1. maximum oil rate;
2. maximum gas rate;
3. maximum water rate;
4. maximum gas-oil ratio;
5. maximum water-oil ratio;
6. maximum wellbore pressure;
7. minimum wellbore pressure;

If a constraint is made negative, it will be ignored. If the well has been previously defined, no more information is required. The next card contains the name of the well (`char*64`)

`wellname`

If it is a new well or if `perf` = 1, the position and geometry of the well must be specified according to the following:

iii) `nlsegs`

`nlsegs` is the number of line segments (element edges) which constitute the well. For each line segment, certain information must be provided (all integer):

iv) `lsegno, lnelem, n1, n2`

`lsegno` is the line segment number; `lnelem` is the number of elements associated with the line segment; `n1` and `n2` are the global node numbers defining the line segment. For each element, the following information is required:

`kk, elem, frac, s11, s12, s21, s22`

`kk` is the count of the current line providing element information for the current line segment, ranging from 1 to `lnelem`; `elem` is the element number in the mesh; `frac` is the fractional contribution of this element to the flow ($\theta/2\pi$); `s11` and `s12` are global node numbers of the support nodes in the upper flow plane, `s21` and `s22` are the node numbers in the lower flow plane. This is illustrated in Figure I–2.

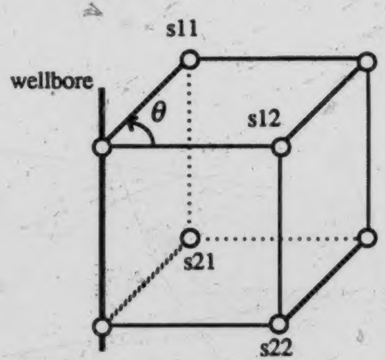


Figure I–2

Well definition in an L8 finite element

`printing data`

Again, the data follow the activation time, which appears on its own line. Only one line of data is required:

`nextout, timeint, stepint, dtnext`

`nextout` is the time at which the simulation information will be written to output (`real`); `timeint` is the simulation time interval between subsequent writes to output (`real`); `stepint` is the interval in time steps between subsequent writes to output (`integer`); `dtnext` is the value to be used for the next time step (`real`). Setting any of these values to zero or negative will cause the program to ignore them.

`control data`

One line of data is required to follow the activation time:

`endtime, endstep, dtmax, rstrtint, rstflag`

`endtime` is the end of the simulation time (`real`); `endstep` is the maximum number of time steps allowed in the simulation (`integer`); `dtmax` is the maximum permitted time step size (`real`); `rsrtint` is the interval in time steps between writing restart records; `rstflag` is set to 1 if restart records are to be written at the same time as output is written. Otherwise set it equal to zero. It is essential that all these data be provided with the correct values. They will never be ignored.

I.4 *Symbolic data set for* FESPS

```
# first character in file is comment character
# wherever it appears in the first column it defines a comment line
This line is the title of the run
strtstep, npts, nelem, mxdofpn, totdof, ngaus
# program will now abort if it needs to be recompiled
# DOF: Sw  So  Pw  T  U  V   W
idof(1), idof(2), idof(3), idof(4), idof(5), idof(6), idof(7)
# now define the units
input units
# some example units (see table in user manual): watch the case.
# putting the description after the key number saves space.
1 - time
days
2 - pressure
kPa
7 - viscosities
mPa.s
13 - well rates
dm3/s
15 - stresses
MPa
# finished defining input units
same units for output units
# self-explanatory
# time step control variables
pnorm, tnorm, satnorm, dispnorm, dampdt, satcon, theta
endtime, dtmax, rstrtint, rstflag
# the required tolerances
tols(1), tols(2), tols(3), tols(4), tols(5), tols(6), tols(7),
tols(8), tols(9)
# the required shifts
shifts(1), shifts(2), shifts(3), shifts(4), shifts(5)
# the required iteration controls
iter(1), iter(2), iter(3), iter(4)
#perturbation values (gradients wrt time) as input unit per day (eg, Pa/day)
perturb(1), perturb(2), perturb(3), perturb(4)
# debugging information
well inputs
well calculations
end debugging
# self-explanatory
#
# the next section deals with initial values of variables
# nodal coordinates: need npts entries, node=1..npts
nodes (data)
refdepth
node, x, y, z
# etc....
```

```
temperatures
# use standard RESIN method for one value satisfying entire range
one
start, end, value
end
#
pressures
equilibrium
datum, pdatum, tdatum, goc, woc
#
sato
one
start, end, value
end
#
satw
one
start, end, value
end
#
porosity
one
start, end, value
end
#
bpp
specified
value
# one value specified for bpp
#
capillary pressure data for oil and water
# Pc_ow = a + bSw* + cSw*^2 + d/(eps + Sw*); Sw* = (Sw - Swi)/(1-Swi)
Swi,  Sor,  epsilon,  a,    b,    c,    d,    m
#
cmod  ### convective multipliers (not required)
water
1.0,  1.0
1.0,  1.0
oil
1.0,  1.0
1.0,  1.0
gas
1.0,  1.0
1.0,  1.0
end of convective multipliers
#
x-permeabilities
ind
start, end
value1, value2, value3
value4, value5
```

```
# etc...
one
start, end
# one can enter as many ranges as one likes
end
ky=x
factor
kz=x
factor
# set all permeabilities equal to x-permeabilities
# end of static definitions
#
# no the well, print, and control data
well data
0.0
# well data to be used at time 0, i.e., the start of the simulation
# define well specifications and constraints
wnum, type, spec, rw, perfs
# wsplit(1), wsplit(2)
# the above two needed if it is an injection well
my well name
con(1), con(2), con(3), con(4), con(5), con(6), con(7)
# if this well has not previously been defined the following data must
# be provided
nlsegs
# need nlsegs of the following groups (lsegno=1..nlsegs)
lsegno, lnelem, n1, n2
# need lnelem cards describing the associated elements (kk=1..lnelem)
kk, elem, frac, s11, s12, s21, s22
# etc....
#
# new type of data
printing data
0.0
nextout, timeint, stepint, dtnext
# new type of data
control data
0.0
endtime, endstep, dtmax, rstrtint, rstflag
# another well specification, active at a later time
well data
100.0
wnum, type, spec, rw
con(1), con(2), con(3), con(4), con(5), con(6), con(7)
#
printing data
100.0
nextout, timeint, stepint, dtnext
#
# we don't need the following, but it informs FESPS that there
# will be no more data to be read (only four dashes actually needed)
```